96769410

CONTROL DATA CORPORATION

MSOS VERSION 5 INSTALLATION HANDBOOK

CDC® COMPUTER SYSTEMS:

CYBER 18
1700

REVISION RECORD

REVISION	DESCRIPTION
01	Preliminary edition
(10/76)	
A	Manual released.
(1/77)	

Publication No. 96769410

REVISION LETTERS I, O, Q AND X ARE NOT USED

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Cover	--
Title Page	--
ii, iii	A
iv	Blank
v	A
vi	Blank
vii, viii	A
1-1	A
2-1, 2-2	A
3-1 thru 3-8	A
4-1 thru 4-7	A
5-1, 5-2	A
6-1 thru 6-15	A
7-1 thru 7-10	A
8-1 thru 8-7	A
9-1 thru 9-4	A
10-1 thru 10-4	A
11-1 thru 11-4	A
12-1	A
13-1, 13-2	A
A-1, A-2	A
B-1, B-2	A
C-1	A
D-1, D-2	A
E-1, E-2	A
F-1	A
G-1, G-2	A
H-1	A
I-1 thru I-6	A
J-1 thru J-29	A
K-1 thru K-8	A
L-1	A
M-1	A
N-1	A
O-1	A
P-1	A
Q-1	A
R-1, R-2	A
S-1 thru S-7	A
T-1 thru T-3	A
U-1 thru U-6	A
V-1	A
Index 1 thru Index 3	A
Comment Sheet	A
Envelope	--
Back Cover	--

PAGE	REV

PAGE	REV

PAGE	REV

PAGE	REV

PREFACE

This installation handbook describes the procedures necessary for the user to install a CDC® CYBER 18/1700 Mass Storage Operating System (MSOS) Version 5 Computer System. It is assumed that release materials have been customized to the particular hardware configuration by the distributing center.

In this manual the term 1700 Computer System refers to any of the following computers:

1704 Computer
1714 Computer
1774 Computer
1784-1 Computer
1784-2 Computer
} CYBER 18-17

The installation procedures are given in terms of release materials being provided on punched cards or magnetic tape (depending on the user's configuration).

It is assumed that the reader has a basic knowledge of the CYBER 18/1700 Mass Storage Operating System.

The sections regarding additions to a system assume the basic system is a standard system. Before reading a section on a given system addition, the user should be familiar with the material in General Procedure for System Additions, section 5.

Additional information may be found in the following publications:

Publication	Publication No.
Small Computer Maintenance Monitor Reference Manual	39520200
File Manager Version 1 Reference Manual	39520600
Macro Assembler Reference Manual	60361900
MS FORTRAN Version 3A/B Reference Manual	60362000
Magnetic Tape Utility Processor Reference Manual	96768400
RPG II Reference Manual	96769000
Sort/Merge Version 1.0 Reference Manual	96769260
Peripheral Drivers Reference Manual	96769390
MSOS Version 5 Reference Manual	96769400
MSOS Version 5 Release Bulletin	96769440
MSOS Version 5 Diagnostic Handbook	96769450
MSOS Version 5 Ordering Bulletin	96769490

This product is intended for use only as described in this document. Control Data Corporation cannot be held responsible for the proper functioning of undescribed features or undefined parameters.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

The installation procedure for Mass Storage Operating System (MSOS) Version 5 has been designed to be as straightforward as possible. The user receives installation material that must be loaded into the system. When this material has been properly loaded and tested, the system is ready for use.

Installation materials are provided either on punched cards, magnetic tape, or a combination of cards and tape, depending on the user's configuration. This material consists of:

- A deadstart card deck (if the system is a CYBER 18-20 with a card reader)
- The system initializer program file
- The installation file
- Two RPG files (if RPG is in the system)
- Three verification files
- A COSY copy of SYSDAT (SYSDAT is a program that contains all the customizable data in the system; by changing SYSDAT, the user can modify the system.)

There is one file mark after the system initializer program and one file mark after the installation file. The user can access the desired part by advancing the proper number of files.

The installation device is the card reader or the magnetic tape unit from which the installation material is read.

Installation proceeds as follows:

- The user loads the system initializer program into main memory. The loading normally requires the loading and executing of a bootstrap, which, in turn, loads the initializer. However, if the user has a current version of MSOS already operating on the equipment, he may use the initializer from the system library.
- The user executes the initializer. This includes reading the initializer parameters and processing the installation data. At the end of this phase, the main memory resident and mass memory resident programs have been loaded and linked, and the number and identity of programs in the system directory have been established. The autoload program is generated at the end of this phase.
- The user autoloads the system and installs the program library. The system is now complete and ready for execution.
- The user verifies the system. After autoloading the system again, the user executes the verification test programs.

Following successful verification, the system is ready to run.

The remainder of this manual amplifies the installation procedures given above and then gives special information on modifying and reinstalling the system in the event that a new product is added to an existing system.

- Section 2 describes hardware requirements for installation.
- Section 3 describes loading and executing the initializer and installing the program library.
- Section 4 describes all verification procedures.
- Sections 5 through 11 describe augmenting an existing system by installing a new product.
- Section 12 describes new features found in the CYBER 18-20 version of MSOS.
- Section 13 describes the procedure for updating MSOS with new installation materials.

The hardware is defined by the requirements of the system; the installation material reflects this configuration. Equipment codes and interrupt lines are standard and are defined in tables 2-1 and 2-2.

Users should be certain the equipment codes and interrupt lines correspond to this equipment configuration.

TABLE 2-1. 1700 COMPUTER SYSTEM

Device Type	Interrupt Line and Equipment Code	Device Type	Interrupt Line and Equipment Code
Low-speed I/O line 1 device	1	1500 Series equipment†	8 and 9
Drum mass memory	2	Card punch	10
1747 Data Set Interface	2	Card reader	11
Disk mass memory	3	1744 Digigraphics Controller	12 and 13
Line printer	4	1745-2 Display Controller	12 and 13
Communications unit	5 and 6	1781-1 Hardware Floating Point Unit	15
Magnetic tape	7		

†The 1590 also uses interrupt line 6.
The 1595 also uses interrupt line 5.
The 1576 also uses interrupt line 15.

TABLE 2-2. CYBER 18-20 SYSTEM HARDWARE REQUIREMENTS

Peripheral	Equipment Code †	Macro Interrupt	Micro Interrupt
Teletypewriter/CRT	1	1	1
Paper tape reader	2	2	2
Paper tape punch	2	2	2
Card punch	2	2	2
None	3	3	3
Line printer	4	4	4
None	5	5	5
None	6	6	6
Tape cassette	7	7	7
Clock	1	8	8

†Equipment codes 0, 3, 5, 6, and 8 are currently unassigned and reserved for future use.

TABLE 2-2. CYBER 18-20 SYSTEM HARDWARE REQUIREMENTS (Contd)

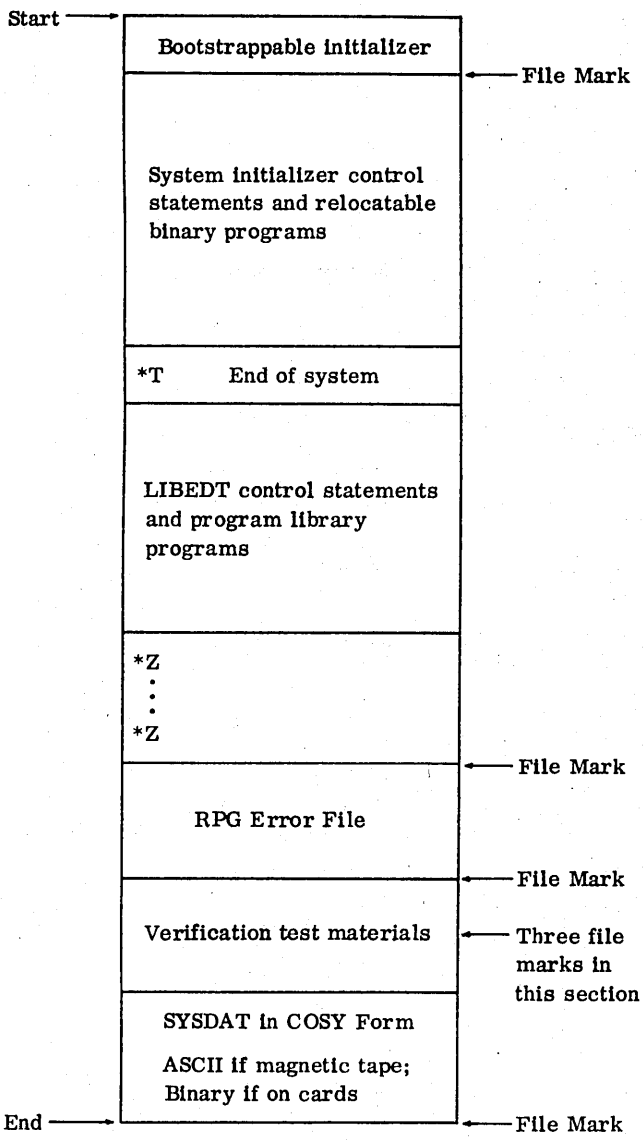
Peripheral	Equipment Code†	Macro Interrupt	Micro Interrupt
1832-4 Magnetic Tape Controller (NRZI only)††	9	9	0 and 9
Eight-channel communications line adapter	10	10	10
Dual-channel communications line adapter	10	10	10
Card reader	11	11	11
1832-4 Magnetic Tape Controller (NRZI and phase encoded)	12	12	N/A
IOM	13	13	N/A
Storage module drive	14	14	N/A
Cartridge disk drive	14	14	N/A
Flexible disk drive	15	15	N/A
Protect, parity, and power failure (internal)	N/A	0	N/A
Macro stop and panel (internal)	N/A	N/A	12-15

†Equipment codes 0, 3, 5, 6, and 8 are currently unassigned and reserved for future use.

††The 1832-4 Magnetic Tape Controller (NRZI only) micro interrupt is wired to both micro interrupt zero and nine. The software has the responsibility to select the desired one.

3.1 SUMMARY

This section describes using the installation materials to build a system. The installation file supplied has a typical format as shown below.



†Supplied only with systems that have RPG.

The installation steps are:

1. The initializer is read into main memory. If the first file of the installation file is used, the loading procedure is described in one of the following sections:
   - 3.2 and 3.4 if using a 1700 Series computer and input materials from a card reader
   - 3.3 and 3.4 if using a 1700 Series computer and input materials from a magnetic tape transport
   - 3.5 if using a CYBER 18-20 Computer and a card reader
   - 3.6 if using a CYBER 18-20 Computer and no card reader
   - 3.7 if using either a 1700 Series or a CYBER 18-20 Computer with a working MSOS system (current MSOS version only).

   At the end of this phase, the initializer is loaded, verified, and ready for execution.
2. The initializer is executed. This is described in section 3.8. Input to this phase is the first part of the system installation file. If this phase is successfully completed, outputs from the phase are:
   - Main memory programs loaded and linked and ready to be autoloaded into the computer
   - System library programs loaded and linked on mass memory, together with a partially completed system library directory
   - Other system programs loaded in mass memory
   - The autoload program

   At the end of the phase, the operator is notified that he may autoload the system.
3. The program library is installed. This is described in section 3.9. Input to this phase is the second part of the system installation file. If this phase is successfully completed, the outputs are:
   - The final system library directory
   - The program library on mass storage, together with a program library directory
   - A complete but unverified MSOS system

At the end of this phase, the operator is notified that he may autoload the system. At this point, the operator normally proceeds to the verification procedures described in section 4.

3.2 CARD READER BOOTSTRAPS (1700 SERIES COMPUTER)

The operator selects the bootstrap below that is associated with his card reader equipment. The bootstrap is entered into main memory, starting at location zero and using the data entry method described in appendix D.

1728-430, 1729-2, OR 1729-3 8-BIT BINARY BOOTSTRAP

Location	Contents
0	0500
1	6823
2	6823
3	E000
4	05A1†
5	C000
6	0081
7	03FE
8	0AD7
9	681A
A	0DFE
B	0B00
C	02FE
D	A815
E	0FC8
F	6C16
10	0B00
11	02FE
12	A810
13	BC12
14	6C11
15	D810
16	0829
17	D80C
18	C80B
19	0121
1A	18F1
1B	C806
1C	086C
1D	0841
1E	0111
1F	1C05
20	18E2
21	0F00
22	00FF
23	0000
24	0000
25	0000

After loading, the operator verifies that the bootstrap code was properly entered (the bootstrap verification method is given in appendix D). Then the operator proceeds to execute the bootstrap by the method described in section 3.4.

1726/405 CARD READER 8-BIT BINARY BOOTSTRAP

Location	Contents
0	0500
1	6821
2	6821
3	E000
4	0581††
5	C81A
6	03FE
7	0DFE
8	0B00
9	02FE
A	A817
B	0FC8

†Use 0521 for 1728-430.
††Use 1581 for 1706 No. 1.

Location	Contents
C	6C17
D	0B00
E	02FE
F	A812
10	BC13
11	6C12
12	D811
13	0829
14	0D01
15	0B00
16	02FE
17	0FCB
18	0125
19	C807
1A	086C
1B	0841
1C	0111
1D	1C05
1E	18E8
1F	0401
20	0F00
21	00FF
22	0000
23	0000

After loading, the operator verifies that the bootstrap was properly entered (the bootstrap verification method is given in appendix D). Then the operator proceeds to execute the bootstrap by the method described in section 3.4.

3.3 MAGNETIC TAPE BOOTSTRAP (1700 SERIES COMPUTER)

The operator selects the bootstrap that is associated with his magnetic tape equipment. After mounting the installation tape on tape drive unit 0, the operator positions the tape to the load point and readies the driver. The bootstrap code is entered into main memory, starting at location zero and using the data entry method described in appendix D.

SEVEN-TRACK MAGNETIC TAPE BOOTSTRAP

Location	Contents
0	0500
1	6824
2	6824
3	E000
4	0382†
5	C81E
6	03FE
7	0DFE
8	C81C
9	03FE
A	0DFE
B	0A00
C	020D
D	0FCA
E	0821
F	0A00
10	02FE
11	0FC4
12	0869
13	0A00
14	02FE
15	0F42
16	086C
17	6C0F
18	D80E
19	18F1
1A	0D01
1B	0B00
1C	02FE
1D	0FCB
1E	0131
1F	18EA
20	C804
21	03FE

†Use 1382 for 1706 No. 1.

Location	Contents
22	1C03
23	0414
24	0100
25	0000
26	0000

After loading, the operator verifies that the bootstrap was properly entered (the bootstrap verification method is given in appendix D). Then the operator proceeds to execute the bootstrap using the method described in paragraph 3.4.

NINE-TRACK MAGNETIC TAPE BOOTSTRAP

Location	Contents
0	6819
1	6819
2	E000
3	0382†
4	C813
5	03FE
6	0DFE
7	C811
8	03FE
9	0DFE
A	0203
B	6C0F
C	D80E
D	18FC
E	0D01
F	0B00
10	02FE
11	0FCB
12	0131
13	18F5
14	C804
15	03FE
16	1C03

†Use 1382 for 1706 No. 1.

Location	Contents
17	044C
18	0100
19	0000
1A	0000

After loading, the operator verifies that the bootstrap was properly entered (the bootstrap verification method is given in appendix D). Then the operator proceeds to execute the bootstrap using the method described in section 3.4.

3.4 BOOTSTRAP EXECUTION (1700 SERIES COMPUTER)

1. Set all switches to neutral.
2. Press MASTER CLEAR.
3. Select the A register.
4. Set the pushbutton register to a value of xxxx, where xxxx is obtained from the following:

Main Memory Size	xxxx
16K	2000
24K	4000
32K	5000
65K	5000

5. Ensure that the initialization material is on the installation device and that the device is ready.
6. Initiate computer execution (GO or RUN). The system initializer is read from the installation device.
7. Execution begins automatically when the system initializer has been completely read.

3.5 CARD READER BOOTSTRAP (CYBER 18-20 COMPUTER)

If there is a card reader in the system, the following deadstart procedure is used to read the bootstrap into macro memory.

1. Press MASTER CLEAR.
2. Place the deadstart program deck in the card reader. If the installation material is on cards, the system initializer must also be in the card reader hopper

feeder. At the end of the system initializer, there is an end-of-file mark as shown in appendix H. If installation material is on tape, mount the tape and load and ready the tape unit.

3. Push the RESET button on the card reader to ready it.

4. Push the DEADSTART button.

5. The bootstrap within the deadstart deck is read into macro memory, and the bootstrap executes automatically. Then the operator proceeds to execute the system initializer by the method described in section 3.8.

The 1700 Series code for the deadstart deck is shown in appendix C.

3.6 MAGNETIC TAPE BOOTSTRAP (CYBER 18-20 COMPUTER)

Mount the installation tape on tape unit 0. Position the tape to the load point and ready the unit.

1. Press MASTER CLEAR.

2. Press ESCAPE to enter panel mode.

3. Enter:

   HG

   This causes a halt.

4. Enter:

   J11G

   This selects the P register.

5. Enter:

   K0000G

   This sets P equal to 0.

6. Enter:

   J07G

   This selects macro memory.

7. Enter:

   LhhhhG, where hhhhG is first line of the appropriate bootstrap from appendix B or C, according to the system installation device.

   This begins loading the bootstrap.

8. Type in the rest of the appropriate bootstrap from appendix B or C. Refer to appendix D for the method of checking the bootstrap.

9. Enter:

   J11G

   This selects the P register.

10. Enter:

    K0000G

    This sets P equal to 0.

11. Enter:

    J14G

    This selects the A register.

12. Enter:

    K5000G

    This sets A equal to 5000.

13. Enter:

    I@

    This starts the system initializer.

Then the operator proceeds to execute the system initializer by the method described in section 3.8.

3.7 INSTALLATION WITH A WORKING MSOS SYSTEM

The system initializer in a working MSOS system may be used to build the new system. Caution must be exercised in using this method, since the system initializer may not be the most current version. Using the system initializer in the installation materials insures that the current version is used.

1. Load the installation material into the proper device. Ready the device if it is a magnetic tape transport.

2. Autoload the system using the method described in appendix F. If using cards, manually remove the first file (i.e., the system initializer program) from the installation material. Then proceed to step 6 below.

3. Enter ODEBUG (tape only).

   Press manual interrupt.

   System responds:

   MI

   Enter:

   DB

System responds:

DEBUG IN

4. Advance one file on the installation material (tape only).

Enter:

ADF, 6, 1

System responds:

NEXT

5. Exit from ODEBUG (tape only).

Enter:

OFF

System responds:

DEBUG OUT

6. Execute the system initializer load program:

MI
*BATCH
*JOB
*SILP

7. The system types:

THE INITIALIZER WILL BE MOVED TO LOCATION xxxx AND EXECUTED TURN OFF PROTEC SWITCH AND TYPE CARRIAGE RETURN

If using a 1700 Series computer system, the operator sets the protect switch to the neutral position and presses carriage return.

If using a CYBER 18-20 Computer system, the operator presses ESCAPE, enters J20@, and presses carriage return. This clears program protect, signals a carriage return, and reverts to operator mode.

Ready the card reader if a card reader is the installation device.

The operator then proceeds to execute the initializer using the method described in section 3.8.

3.8 SYSTEM INITIALIZER EXECUTION

1. When the system initializer begins execution, the following messages are output on the console:

MSOS 5.0 SYSTEM INITIALIZER
FWA ADDRESS OF CONTROL = xxxx

The value of xxxx is the starting address of the system initializer and may be used to restart the initializer if necessary.

2. If the system is to be built on a disk, ensure that the required disk pack is mounted on unit 0.

3. The system outputs the following message:

DATE MM/DD/YY

Enter the date in the form of month/day/year (two digits each).

4. The normal list device for the initializer is the console display or teletypewriter. If output is desired on the printer, enter:

*C, 7

5. If the system is being installed on a new disk pack, address tags may be written on the pack by using the *G control statement. In a CYBER 18-20 computer system using an 1833-1 Storage Module Drive, the *G control statement causes data to be written on the entire disk as well as on address tags (see appendix G). Surface tests may be run on a new pack by using the *H control statement. (This requires several hours.)

6. If loading from cards, enter:

*I, 2

If loading from tape, enter:

*I, 3

7. If loading from cards, empty the output card hopper of any cards that have been read and load more cards into the input hopper.

8. Enter:

*V

This command instructs the initializer to begin reading control statements from the load device. As the installable binaries are read, the program names are typed out on the list device in the form:

name xxxx identification summary level

Where: name is the name of the program.

identification is the program deck identification.

xxxx is one of the following:

- The first word address (FWA) of the program if the program is main memory resident (*L or *LP)
- The beginning sector number of the first program of a group of programs associated with a *YM ordinal for mass memory resident

programs (*M or *MP). These are system library programs.

- The relative address of a program within a *YM ordinal when that program is not the first program in the ordinal

9. A sample of the initialization printout is given in appendixes I and J. The printout differs for individual systems depending on the configurations and options used.

10. There are three distinct pauses during the execution of the initializer:

    - After all *L statements have been read from the installation file
    - After all *LP statements have been read from the installation file
    - At the conclusion of the installation

    These pauses may be as long as 30 seconds; they occur because two internal tables are being generated at this point (CREP and CREP1).

11. If installing from cards, it is periodically necessary to empty the output card hopper and load more cards into the input hopper. If using a line printer as the list device, a convenient way of halting the initializer while removing and loading cards is to press the ready pushbutton on the line printer, causing the ready indicator light to go off. This halts the initializer. After removing and loading the cards, again press the ready pushbutton. The ready indicator light illuminates, and the initializer resumes operation. The ready pushbutton on the card reader should not be pressed to halt operation, since this may cause errors. If using a CYBER 18 computer, the loading and removal of cards may be done in the pauses between card reading without halting the system. If the list device is not a line printer, the slower speed of output to the comment device allows card removal and loading without halting the system.

12. If errors occur, error messages are output in the form:

    ERROR x.

    The values of x are given in appendix E.

13. The following system externals appear as unpatched at the conclusion of the system build if related routines are not included in the system:

    PARITY — Entry point of a user-supplied program to process core parity errors

    POWERU — Entry point of a user-supplied program to restart after a power failure

    In addition the following externals appear as unpatched in 1700 Series systems:

    SRG721 — Entry point of a user-supplied routine to handle 1572-1 Sample Rate Generator interrupts

    LST721 — Entry point of a user-supplied routine to handle 1572-1 line sync timer interrupts

    STALLD — Entry point of a user-supplied routine to handle 1576-1 Stall Alarm Unit interrupts

    Any other unpatched externals should be considered errors, and their cause should be investigated.

14. At the conclusion of the installation, the system outputs either of the following messages:

    INITIALIZATION COMPLETE — YOU MAY AUTOLOAD

    or

    ERRORS OCCURRED — YOU MAY ATTEMPT TO AUTOLOAD

    If the latter message occurs, the significance of the error messages output should be checked.

3.9 PROGRAM LIBRARY INSTALLATION

1. Autoload the system (see appendix F).
2. Press manual interrupt.
3. The system outputs:

    MI

4. Enter:

    *BATCH

    The system library priorities are set and the program library is not built. See appendix I for a sample listing of the load map.

5. If installation is made using cards, it is periodically necessary to stop the job to empty and to load the respective card reader hopper feeders. The steps that accomplish this operation are:

    a. Press manual interrupt.

    b. The card reader stops reading cards, and the system outputs MI on the console display.

c. Empty the output hopper and load more cards into the input hopper.

d. When ready to continue, press carriage return. The loading continues immediately.

6. The following messages are printed if, and only if, RPG is included in the system:

```
RPG II DISK FILES WILL BE INITIALIZED
RPG II DISK FILES INITIALIZED
RPG II ERROR MESSAGE FILE IS LOADED
```

7. At the conclusion of installation, the following message is output:

```
*CTO, MSOS 5.0 INSTALLATION COMPLETED — YOU MAY AUTOLOAD
```

8. Autoload the system.

9. At this point, the installation material is positioned at the beginning of the MSOS verification test materials, and these tests should now be executed. Section 4 describes these tests and their operation.

4.1 TEST OPERATION SUMMARY

Verification tests are loaded as follows:

1. Place the verification test materials in the appropriate input device. Ready the unit. The installation materials are properly positioned and ready if the system has just been installed.
2. Ready the system list device.
3. If the tests are not being run at the conclusion of a system build, advance past the proper number of files of the installation material to locate the verification tests (see section 1 and figure 3-1).
4. Press manual interrupt.
5. The system outputs:

   MI

6. Enter:

   VERIFY

The verification tests run to completion without further operator intervention.

4.2 TEST DESCRIPTION

The MSOS verification tests are a completely automated set of tests that exercise the major elements of the installed system. Operator intervention is not required during execution; successful passage signifies a correctly installed operating system.

The tests are controlled by an executive named VERIFY that resides in the system library ordinal. Tests that allow verification of the complete set of MSOS elements are always supplied, and the executive selects and sequences the tests required by the installed configuration. In this way, elements may be added to an existing installation, and their correct operation may be easily verified.

The test executive utilizes a pseudo driver for the system comment and listing devices during some portions of the tests. The pseudo comment driver does not perform input/output operations, but it does trap certain system messages and allows transfer of control between the system background and foreground. The pseudo listing driver causes listing records to be written to an area of mass storage temporarily allocated in system scratch. These records are used during the verification of the macro assembler, FORTRAN compiler, RPG compiler, and several other system elements. To avoid interference with the remainder of the system, the pseudo driver resides in the system communications region (locations 47_{16} through $B2_{16}$).

None of the system input/output drivers are specifically exercised as a part of the verification tests; however, the drivers for the system comment, list, installation, and library units are indirectly exercised during the test operation.

Because the verification materials consist of mixed ASCII and binary information, they cannot be copied either from or to a seven-track magnetic tape transport under MSOS.

4.2.1 REQUIREMENTS

The verification tests are normally used to verify a newly installed or updated system, and the tests require certain standard MSOS features. Highly customized MSOS systems may not allow execution of the verification tests.

4.2.2 SYSTEM TIMER

The system hardware timer or software pseudo timer must be operational for proper execution of the verification test executive.

4.2.3 SYSTEM COMMUNICATIONS REGION

The verification tests make use of the entire communications region between locations 47_{16} and $B2_{16}$. This area is restored from the system core image at the normal conclusion of the tests, but no data reference or other program execution may occur in this region during test execution.

4.2.4 STANDARD LOGICAL UNITS

The MSOS standard logical unit assignment must exist in a system that is verified. Consult the MSOS Reference Manual for a description of this standard. In particular, units must be assigned as follows.

- Logical unit 2 — Dummy device
- Logical unit 4 — Comment device
- Logical unit 9 — List device
- Logical unit 10 — Input device
- Logical unit 12 — FORTRAN list device

If a pseudo tape test is required, logical unit 7 must be the unit 0 pseudo tape. Similarly, if a magnetic tape simulator test is required, logical unit 7 must be the unit 0 simulated magnetic tape. Either of these tests is omitted without error if logical unit 7 is not the specified device.

4.2.5 TEST EXECUTIVE ORDINAL

The system library entry used for the verification tests must be assigned to ordinal 25. This entry is specified by the name VERIFY in the system initializer *YM declarations.

4.2.6 VERIFICATION LOGICAL UNIT

The materials required during the execution of the verification tests are read from the logical unit used during system installation. At the start of the tests, a check is made to ensure that the materials reside on this unit, and the message:

TEST MATERIALS NOT LOADED

appears on the system comment device if this is not the case. Appendix Q contains an illustration of the verification materials.

4.2.7 RESERVED FILES

If the system contains an MSOS file manager, files with hexadecimal numbers 7FFD, 7FFE, and 7FFF are used during the file manager verification tests. If the system contains pseudo tape job files, then the following file names are used during the verification tests:

Job File	Security Code
RPFIL1	RP1111
RPFIL2	RP2222
RPFIL3	RP3333

These files and file names should not be used by applications programs, since any data contained in them is destroyed by the tests.

4.2.8 MSOS ELEMENT COMPONENTS

If the system contains a FORTRAN compiler, a verification of the background FORTRAN library is performed in addition to the compiler verification test.

It is assumed that the library contains all components that comprise the released single-precision nonre-entrant FORTRAN library, as indicated in section 6.2. If the double-precision library is present and if the length of unprotected memory is greater than 9400 decimal words, this library is tested separately. If either of these two conditions does not occur, the test is omitted without error.

If the system contains the re-entrant FORTRAN library, tests are performed on all single-precision library components. These tests assume that priority levels 4 and 5 are re-entrant FORTRAN levels. If present, the re-entrant double-precision library is tested separately.

Removal of programs or components from either FORTRAN library may cause erroneous test results. RPG II, RPG II routine, and Sort/Merge are also tested if they are present in the system.

4.3 TEST OPERATION

The verification tests are initiated by pressing manual interrupt and entering the mnemonic VERIFY. If the system has just been built from the installation file, the verification materials are properly positioned on the installation device and the tests begin by printing the following message on the system comment device:

MSOS 5 VERIFICATION TESTS — PSR LEVEL nnn

Where: nnn is the PSR summary level that is compatible with the tests.

Erroneous results may occur if the value of nnn disagrees with the summary level of the system.

If the tests are not run as a part of system installation, the materials may be properly positioned by:

- Advancing the proper number of files if the medium is magnetic tape
- Advancing the proper number of files or visually locating the verification file if the medium is punched cards

Refer to section 1 and figure 3-1.

Several pauses should be expected during test execution. These may result from running the tests or from loading the test programs. However, a pause of over 5 minutes without apparent system activity should be considered a test error.

The verification tests are divided functionally into 15 sections, five of which are required. The remainder of the sections are selected by the test executive only if the optional system element is present. Each section is initiated by a message on the system comment device specifying the tested element; each section is concluded with a message of similar format. Within each section, a message is output as a test is initiated, and if the test is successful, the following message is output:

-VERIFIED

An example of the output from a successful set of tests is shown in figure 4-1.

Successful completion of the verification tests is evidenced by the message:

MSOS VERIFICATION TESTS COMPLETE.

on the system comment device. At this point, the system is in a normal condition and ready for use.

4.4 INDIVIDUAL TEST DESCRIPTIONS

4.4.1 DIRECTORY LISTING

This is a required test that causes the system library and program library directories to be output on the system list device. The format of the listing is shown in appendix K, but individual addresses and entries vary depending on the system configuration.

4.4.2 LOGICAL UNIT LISTING

This is a required test that causes the system logical units to be listed on the system list device in the format shown in appendix K. Individual entries vary, based on the system configuration.

4.4.3 MONITOR TEST

This is a required test that verifies those areas of the MSOS monitor not exercised during the installation of the system and program libraries.

The program schedule request verification involves execution of several priority levels in proper sequence and the execution resulting from a burst of schedule requests in proper sequence.

The timer request verification establishes correct time delays for various delay units, as well as providing an external measured delay that can be calibrated to a wall clock.

The memory space and release requests are verified by oversubscribing allocatable core with requests totaling 40K words. Release requests are performed in each allocated block after a time delay to allow the test to complete.

The verification of the directory schedule, enable-schedule, and disable-schedule involves periodic scheduling of the verification ordinal, during which a disable-schedule request is made. This is followed by an enable-schedule request to complete the test.

The partition core test is an optional part of the monitor verification. It involves the use of the allocation and the release of partitioned memory. In addition, some checks are made to ensure that the partitions are properly set up.

4.4.4 FILE MANAGER TEST

This is an optional test that verifies correct operation of the MSOS File Manager Version 1.

Sequential files are verified by writing and reading records sequentially in a file. File locking and direct storage and retrieval are also tested.

Simple indexed files are verified by writing and reading indexed records in an indexed file. File locking and locked record retrieval are also tested.

Indexed-ordered files are verified by storing and retrieving records that are ordered by a key value and ensuring that each record contains correct data.

Indexed-linked files are verified by storing and retrieving records that are linked in a first-in, first-out basis by the same key value. Various key values are used in this test.

4.4.5 PSEUDO TAPE TEST

This is an optional test that verifies correct operation of the pseudo magnetic tape driver. The test consists of

```
MI
VERIFY
MSOS 5 VERIFICATION TESTS -- PSR LEVEL 110

THE SYSTEM AND PROGRAM LIBRARY DIRECTORIES
WILL BE LISTED ON THE SYSTEM LIST DEVICE

THE SYSTEM LOGICAL UNITS WILL BE
LISTED ON THE SYSTEM LIST DEVICE

BEGIN MSOS MONITOR TEST
 PROGRAM SCHEDULE REQUEST    -VERIFIED
 TIMER DELAY REQUEST         -VERIFIED
 MEASURED DELAY (30 SEC.)    -VERIFIED
 MEMORY SPACE REQUEST        -VERIFIED
 SPACE RELEASE REQUEST       -VERIFIED
 DIRECTORY SCHEDULE REQUEST  -VERIFIED
 DISABLE - SCHEDULE REQUEST  -VERIFIED
 ENABLE - SCHEDULE REQUEST   -VERIFIED
 PARTITIONED CORE REQUEST    -VERIFIED
MSOS MONITOR TEST COMPLETE

BEGIN MSOS FILE MANAGER TEST
 SEQUENTIAL FILES            -VERIFIED
 INDEXED FILES               -VERIFIED
 INDEXED ORDERED FILES       -VERIFIED
 INDEXED LINKED FILES        -VERIFIED
MSOS FILE MANAGER TEST COMPLETE

BEGIN PSEUDO TAPE TEST
 FOREGROUND REQUEST          -VERIFIED
 JOB FILE REQUEST            -VERIFIED
PSEUDO TAPE TEST COMPLETE

BEGIN MACRO ASSEMBLER TEST
 ASSEMBLER EXECUTION         -VERIFIED
 PROGRAM LISTING             -VERIFIED
 PROGRAM CROSS-REFERENCE     -VERIFIED
 PROGRAM BINARY OUTPUT       -VERIFIED
MACRO ASSEMBLER TEST COMPLETE

BEGIN LIBRARY BUILDER TEST
 LIBRARY BUILDER EXECUTION   -VERIFIED
 LIBRARY BUILDER OUTPUT      -VERIFIED
LIBRARY BUILDER TEST COMPLETE

BEGIN FORTRAN COMPILER TEST
 COMPILER EXECUTION          -VERIFIED
 PROGRAM LISTING             -VERIFIED
 PROGRAM BINARY OUTPUT       -VERIFIED
FORTRAN COMPILER TEST COMPLETE

BEGIN FORTRAN LIBRARY TEST
 FORTRAN EXECUTION           -VERIFIED
 FORMATTED INPUT-OUTPUT      -VERIFIED
 ENCODE-DECODE               -VERIFIED
 ARITHMETIC LIBRARY          -VERIFIED
 MONITOR INTERFACE           -VERIFIED
FORTRAN LIBRARY TEST COMPLETE

BEGIN DOUBLE-PRECISION TEST
 FORTRAN EXECUTION           -VERIFIED
 FORMATTED INPUT-OUTPUT      -VERIFIED
 ENCODE-DECODE               -VERIFIED
 ARITHMETIC LIBRARY          -VERIFIED
DOUBLE-PRECISION TEST COMPLETE

BEGIN RE-ENTRANT FORTRAN TEST
 FORTRAN EXECUTION           -VERIFIED
 FORMATTED INPUT-OUTPUT      -VERIFIED
 ARITHMETIC LIBRARY          -VERIFIED
 MONITOR INTERFACE           -VERIFIED
RE-ENTRANT FORTRAN TEST COMPLETE

BEGIN DOUBLE-PRECISION TEST
 FORTRAN EXECUTION           -VERIFIED
 FORMATTED INPUT-OUTPUT      -VERIFIED
 ARITHMETIC LIBRARY          -VERIFIED
DOUBLE-PRECISION TEST COMPLETE

BEGIN RPG COMPILER VERIFICATION
 RPG COMPILER EXECUTION      -VERIFIED
 PROGRAM LISTING             -VERIFIED
 PROGRAM CROSS-REFERENCE     -VERIFIED
 PROGRAM BINARY OUTPUT       -VERIFIED
RPG COMPILER TEST COMPLETE

BEGIN RPG RUNTIME VERIFICATION
 RUNTIME EXECUTION           -VERIFIED
 RPG RUNTIME OUTPUT LISTING  -VERIFIED
RPG RUNTIME COMPLETE

BEGIN SORT/MERGE VERIFICATION
 SORT/MERGE EXECUTION        -VERIFIED
 SORT/MERGE OUTPUT LISTING   -VERIFIED
SORT/MERGE TEST COMPLETE

MSOS VERIFICATION TESTS COMPLETE
```

Figure 4-1. Verification Test Output Example

two optional parts that depend on the type of pseudo tapes contained in the system.

Foreground pseudo tape requests are verified by performing formatted and unformatted reads and writes to the pseudo tape unit. In addition, all tape motion commands are exercised as a part of the test.

The job file requests are verified in a similar manner by the use of formatted and unformatted read and write requests, as well as all tape motion commands. In addition, the MSOS job processor statements that are required by the job file pseudo tape (e.g., *DEFINE, *RELEASE, *OPEN, etc.) are exercised as a part of the test.

4.4.6 MAGNETIC TAPE SIMULATOR TEST

This is an optional test that verifies correct operation of the magnetic tape simulator. The magnetic tape simulator is verified by the use of formatted and unformatted read and write requests, as well as by the use of all tape motion commands.

4.4.7 MACRO ASSEMBLER TEST

This is a required test that verifies correct operation of the MSOS Macro Assembler Version 3.

The assembler execution is verified by causing a test program to be assembled with the listing directed to the test executive pseudo listing driver and the binary object code to the MSOS load-and-go file. Following assembler execution, the listing and binary data generated during the test are compared with the verification data contained on the verification logical unit. This test is successful if all data agree.

A listing of the program used to test the assembler may be found in appendix S. This program is not designed to be executed.

4.4.8 LIBRARY BUILDER TEST

This is a required test that verifies the correct operation of the MSOS Library Builder utility (LIBILD).

Library builder execution is verified by causing LIBILD to be run, using the verification logical unit as input and the pseudo listing device as output. Following execution, the data generated by LIBILD is compared to a set of data contained in the verification logical unit.

4.4.9 FORTRAN COMPILER TEST

This is an optional test that verifies the correct operation of the MSOS FORTRAN Compiler Version 3.3.

Compiler execution is verified in the same manner as the macro assembler, using the pseudo listing device and the MSOS load-and-go file. A set of listing and binary comparison data is contained in the verification logical unit for both the A and B compiler variants, since each produces slightly different codes from the same source statements.

A listing of the programs used to test the FORTRAN compiler may be found in appendix T. These programs are not designed to be executed.

4.4.10 FORTRAN LIBRARY TEST

This is an optional test that verifies the correct operation of the single-precision, nonre-entrant FORTRAN runtime library, utilizing either the software or hardware floating point unit, depending on which is in the system.

This test is divided into an execution phase and a verification phase. During execution, several programs are run that exercise various portions of the FORTRAN run-time library and write data on the pseudo listing device. Following this, the verification phase is entered in which this data is compared to data contained in the verification logical unit.

The formatted input/output test involves the use of the FORTRAN input/output library by reading and writing data using numerous formats. Unformatted FORTRAN file input/output is also tested.

The encode-decode test involves the use of the FORTRAN encode-decode library as well as additional FORTRAN formatting routines.

The arithmetic library test exercises all of the intrinsic and external functions contained in the run-time library.

The monitor interface test verifies that the nonre-entrant FORTRAN monitor interface is operating properly by exercising such modules as LINK, READ, WRITE, TIMER, etc.

4.4.11 DOUBLE-PRECISION TEST

This is an optional test that verifies the correct operation of the nonre-entrant double-precision FORTRAN run-time library and is structured similarly to the single-precision library test. All double-precision options contained in the formatted input-output, encode-decode, and arithmetic libraries are verified. Depending on the system, this test is performed with either the software or hardware floating point.

4.4.12 RE-ENTRANT FORTRAN LIBRARY TEST

This is an optional test that verifies the correct position of the single-precision, re-entrant FORTRAN run-time library and is structured similarly to the nonre-entrant test. Depending on the system, this test is performed with either the software or hardware floating point.

In addition to testing the formatted input-output, arithmetic, and monitor interface libraries, multi-programming in FORTRAN at priority levels 4 and 5 is also verified.

4.4.13 RE-ENTRANT DOUBLE-PRECISION TEST

This is an optional test that verifies the correct operation of the re-entrant double-precision FORTRAN library and is equivalent to the nonre-entrant test. Depending on the system, this test is performed with either the software or hardware floating point.

4.4.14 RPG II COMPILER TEST

This is an optional test that verifies the correct operation of the RPG II Compiler, Version 1.0. A listing of the program used is contained in appendix U. Compiler execution is verified in the same manner as for the macro assembler, using the pseudo listing device and the MSOS load-and-go file.

4.4.15 RPG II RUNTIME TEST

This is an optional test that verifies the correct operation of the RPG II Version 1.0 runtime library. This section is divided into an execution phase and a verification phase.

4.4.16 SORT/MERGE TEST

This is an optional test that verifies the correct operation of the Sort/Merge package. This test verifies the sort function.

4.5 ERROR CONDITIONS

Verification test errors may be divided into two categories: operating system errors and verification errors. Operating system errors are indicated by hardware input-output errors, protect violations from the background, etc. Refer to the MSOS Reference Manual for a summary of these errors. All operating system errors (including input-output errors resulting from devices in a nonready condition) are fatal to the verification tests. No attempt should be made to continue after an operating system error has occurred.

Verification errors are always evidenced by one of the error messages contained in appendix R and are, in general, recoverable.

Errors found during the monitor tests result in termination of the verification tests. Errors found during the remaining tests result in the termination of that section of the test but the remainder of the test continues.

4.5.1 ERROR MODE

The verification tests may be run in error mode to aid in the isolation and correction of errors. Error mode is enabled as follows:

- If using a 1700 Series computer system, set the selective skip switch UP.
- If using a CYBER 18-20 computer, press ESCAPE and type:

  J20@

  (This sets selective skip and reverts to operator mode.)

When error mode has been enabled, the following actions occur:

- The message:

  ERROR MODE SELECTED

  appears immediately following the initial test message.
- The system list device is not disabled during the loading of individual tests. This can be helpful if test loading results in unpatched externals.
- If a verification error occurs, the system halts rather than contining with the next test. This allows the execution of a core dump or the system checkout bootstrap. Refer to the MSOS Reference Manual for an explanation of the tools.

It should be pointed out that the verification tests perform successfully with error mode selected as long as no errors occur.

4.5.2 ERROR RECOVERY

The verification tests are designed to allow the tests to be restarted in case of an error. Although most of the verification errors result in a continuation of the tests until they are completed, it is advisable to autoload the system before restarting the tests. Autoloading is required if an operating system error has occurred. The following steps should be performed when restarting the tests.

1. Autoload the system.
2. Position the verification materials properly in the installation logical unit.
    - Magnetic tape — Backspace one file using ODEBUG. If the verification error occurred within RPG runtime, backspacing two files may be necessary. If the verification error occurred during Sort/Merge, backspacing three files may be required (see figure 3-1).
    - Punched cards — Visually locate the file mark that precedes the verification materials and load them in the reader.
3. Enter MI and VERIFY to restart the tests.

4.5.3 ERROR MESSAGES

All verification error messages are preceded by three asterisks. File manager error messages include the value of the file manager status word (REQIND). Refer to the File Manager Reference Manual for a description of the status bits. Pseudo tape error messages include the value of the V-field and the driver status. Refer to the MSOS reference manual for a description of these terms. Macro assembler, FORTRAN compiler, and RGP compiler error messages include the line number and the expected line of data for listing errors and the type of loader block (NAM, RBD, etc.) for binary data errors. Refer to the MSOS Reference Manual for a description of loader blocks. Appendix R contains a complete list of these error messages.

This section describes two general procedures for adding a product to CYBER 18/1700 MSOS. Method 1 produces a new installation file and can be used in all cases. Method 2 can be used only on those products that do not require changes to the data base, SYSDAT. The output of this method is a new installation file solely for the product to be added. This new file is not merged with the old installation file. The new product is then installed in the system using the LIBEDT utility; reinstallation of the entire system is not necessary. Method 2 has the advantage of being quick and easy; the disadvantage is that if the system is reinstalled at a later date, reinstallation requires two steps:

- Installing the original installation file.
- Installing the new product using LIBEDT.

Table 5-1 shows the six products that may be added to MSOS, the methods available to install the products, and the manual section where the installation procedure is described in detail.

When one of the above products is purchased as an addition to an existing MSOS, the user receives a binary copy of each module needed to install the ordered product. A compressed source (COSY) copy of the product (and any applicable COSY correction card images) are sent to the user only if these are specifically requested. In a COSY copy of a product, each module is preceded by a COSY identifier.

The utilities used (COSY, SKED, LIBILD, and LIBEDT) are discussed in detail in the MSOS Reference Manual.

TABLE 5-1. SYSTEM ADDITIONS

Section	Product	Installation Method
6	FORTRAN Version 3A/B	1 only
7	File Manager Version 1.0	1 only
8	Report Program Generator (RPG II) Version 1.0	1 only
9	Macro Assembler Version 3.0	1 or 2
10	Sort/Merge Version 1.0	1 or 2
11	Magnetic Tape Utility Processor (MTUP) Version 2.0	1 or 2

5.1 INSTALLATION USING METHOD 1

For method 1, three elements are involved in the addition of the new product:

- The binary copy of the new product
- An existing system installation file. The file is summarized in figure 3-1; it consists of binary object records of each module in a particular MSOS installation, together with system initializer control statements and LIBEDT control statements. The system installation file provides the information needed to build a particular CYBER 18/1700 MSOS system. This information consists of two parts:

  -The first part is used by the system initializer and is shown in appendix I.

  -The second part is used by the library editor (LIBEDT) and is shown in appendix J.

  An installation file may exist on cards, on magnetic tape, or on a combination of both.
- The current version of SYSDAT. This program contains the parameters defining the system. Appropriate parameters must be altered when adding the product. This is accomplished by altering SYSDAT and replacing the object program on the existing installation file.

The binary programs of the new product are merged with the existing installation file, which has been modified to reflect the new SYSDAT configuration. This merged installation tape is then installed and verified using the procedures of sections 3 and 4.

Table 5-2 summarizes the addition of a product using methods 1.

5.2 INSTALLATION USING METHOD 2

The only elements involved in addiging a new product by method 2 are the skeleton records and the binary copy of the product. Table 5-3 summarizes the addition of a product using method 2.

TABLE 5-2. INSTALLATION OF A NEW PRODUCT USING METHOD 1

Task	Materials Used	Methods Including MSOS Packages That May Be Used	Result
Modify SYSDAT	SYSDAT source in COSY form	COSY may be used to make corrections to SYSDAT. If using cards, COSY corrections may be punched on cards with proper COSY control cards. If tape is used, SYSDAT may be modified by using COSY and making COSY corrections from the comment device.	Modified SYSDAT source
Produce SYSDAT binary	Modified SYSDAT source expanded from COSY or modified COSY of SYSDAT	Macro assembler	Binary copy of modified SYSDAT
Generate a new skeleton	Installation file	SKED — Manual revisions of skeleton if on cards	Modified system skeleton
Generate a new installation file	a. Modified system skeleton b. Binaries for product to be added c. Binary copy of modified SYSDAT	LIBILD	Modified installation
Rebuild the system	Modified installation file	System initializer	Modified MSOS on mass memory
Verify the system	Verify new system	VERIFY program operating in background	Modified and verified MSOS on mass storage

TABLE 5-3. INSTALLATION OF A NEW PRODUCT USING METHOD 2

Task	Materials Used	Methods Including MSOS Packages That May Be Used	Result
Produce skeleton records for the installation file	Installation materials for skeleton records	Prepare the file using SKED.	File of sequential records
Create an installation file for the new product	Skeleton file just produced	Construct the new installation file using LIBILD.	New installation file (for new product only)
Install the new product on the system	Installation file just produced	Enter the new product on the program library using LIBEDT.	Modified MSOS on mass memory
Verify the new product	Verification materials on old installation file	VERIFY program	Modified and verified MSOS on mass storage

A user who does not have FORTRAN in the MSOS originally ordered from Control Data may add either the FORTRAN Version 3.3A Compiler or the FORTRAN Version 3.3B Compiler to his system. A description of the two compilers may be found in the MS FORTRAN Version 3A/B Reference Manual. To add FORTRAN, the user must first order the installation materials for the compiler desired (refer to the MSOS 5 Ordering Bulletin). The FORTRAN installation materials sent to the user are FORTRAN binaries (on magnetic tape or cards) including either the FORTRAN Version 3.3A Compiler or the FORTRAN Version 3.3B Compiler together with:

- Each module of the FORTRAN re-entrant library
- Each module of the FORTRAN nonre-entrant library

The user must modify SYSDAT and generate a new skeleton. A new installation file is then created using LIBILD, and a new system is built.

6.1 SYSDAT MODIFICATIONS

Modifications to the SYSDAT program are required if the system is to contain either the re-entrant FORTRAN library or a 1781-1 Hardware Floating Point Unit. After determining and making the required modifications, SYSDAT must be reassembled. This new version replaces the SYSDAT program in the system installation file.

6.1.1 SYSDAT MODIFICATIONS NECESSARY TO ADD RE-ENTRANT FORTRAN

In the storage stacks section of SYSDAT, change the definition of NFTNLV and NEDLVL as follows:

```
NFTNLV EQU NFTNLV (n) NUMBER OF
REENTRANT FORTRAN LEVELS

NEDLVL EQU NEDLVL (n) NUMBER OF
REENTRANT ENCODE/DECODE LEVELS
```

Where: n is the number of re-entrant FORTRAN levels to be in the system.

The normal FORTRAN levels are 4, 5, and 6 with n equal to 3.

In the miscellaneous information section of SYSDAT, delete the following code:

```
      SPC  4
*     THIS ENTRY IS PROVIDED TO LINK THE
      FORTRAN REENTRANCY DATA
*     ENTRY POINTS
      SPC  1
      ENT  FMASK, FLIST, DOUT
      SPC  1
      EQU  FMASK($7FFF), FLIST($7FFF), DOUT
       ($7FFF)
```

Replace this code with the code in figure 6-1.

6.1.2 SYSDAT MODIFICATIONS NECESSARY WHEN ADDING FORTRAN TO SYSTEM WITH 1781-1

Add the following code at any convenient point in SYSDAT after the physical device tables:

```
*     MISCELLANEOUS INFORMATION
*     1781-1 HARDWARE FLOATING
*     POINT INFORMATION
*
*

          ENT  E17811  CONVERTER, EQUIPMENT,
                       STATION

E17811    NUM  *0783   EQUIPMENT CODE = 15

          ENT  F17811  INITIAL FUNCTION TO SET
                       OPERATING MODE

F17811    NUM  $0000
```

Add the following preset to the table of presets at the end of SYSDAT:

```
*          1781-1 EQUIPMENT CODE PRESET

      ALF  3, E17811

      ADC  E17811
```

```
          EJT
*              MISCELLANEOUS INFORMATION
*
*              FORTRAN REENTRANT INFORMATION
*
          ENT   FMASK, FLIST
          EXT   E4SAVE
          EXT   ARGU0
          SPC   1
FMASK     NUM   $0070              FORTRAN REENTRANT LEVELS (BIT 0 = LEVEL 0)
*         TABLE OF FORTRAN ENTRY POINTS SAVED TO MAINTAIN REENTRANCY
          SPC   1
*         ENTRY POINT              PROGRAM              DESCRIPTION
*         -------------            ----------           -------------
          SPC   1
FLIST     ADC   FEND
          ADC   E4SAVE             Q8EXPR          LOCATION $E4   STORAGE
          ADC   ARGU0              Q8GTO           TEMPORARY STORAGE
FEND      EQU   FEND(*-FLTST-1)
          EJT
*              FORTRAN REENTRANT INFORMATION
*
          SPC   4
*         THIS ENTRY IS PROVIDED TO ALLOW COMPATIBILITY BETWEEN THE
*         NON-REENTRANT (BACKGROUND) FORTRAN AND REENTRANT FORTRAN
          SPC   1
          ENT   Q8STP
          SPC   1
Q8STP     NCP   0
          JMP-  (ADISP)
```

Figure 6-1. Re-entrant FORTRAN Table

Delete the following code from the miscellaneous information section of SYSDAT:

```
          EJT
*              MISCELLANEOUS INFORMATION
          SPC   4
*         THESE ENTRIES ALLOW PROPER
          SYSTEM LINKAGE
          SPC   2
          ENT   E17811, E17811
E17811    NUM   $7FFF
E17811    NUM   $7FFF
          EJT
```

If double precision is not to be included, insert the following code into the miscellaneous information section of SYSDAT:

```
          SPC   2
*         LINK THE DOUBLE PRECISION ENTRY
          POINT REFERENCED BY 'FORMTR'
          SPC   1
          ENT   DOUT
          EQU   DOUT($7FFF)
```

In this section of SYSDAT which includes the COBOP starting sector, insert the following code:

```
          SPC   4
*         THIS ENTRY IS PROVIDED TO LINK THE
          NO-FORTRAN DISPATCHER
*         ENTRY POINT
          SPC   1
          ENT   NDTSP
          SPC   1
          EQU   NDTSP($7FFF)
```

6.2 SYSTEM SKELETON MODIFICATION

The current system skeleton must first be obtained (refer to appendix N). If the skeleton is obtained on cards, it may be manually modified. If it is obtained on tape, the utility system skeleton editor, SKED, may be used to perform the modifications.

If the re-entrant FORTRAN is being added, replace the appropriate record:

```
*B 'NDISP' ' DECK-ID 059   MSOS 5.0'
   (CYBER 18-20 Computer)
*B 'NDISP' ' DECK-ID M24   MSOS 5.0'
   (1700 Series Computer)
```

with one of these records:

```
*B 'RDISP' ' DECK-ID 058   MSOS 5.0'
   (CYBER 18-20 Computer)
*B 'RDISP' ' DECK-ID M23   MSOS 5.0'
   (1700 Series Computer)
```

In addition, when adding re-entrant FORTRAN, insert the re-entrant FORTRAN runtime library skeleton records immediately before the record:

```
*B 'NXTLOC' ' NEXT AVAILABLE LOCATION'
```

which precedes the system mass resident programs.

NOTE

The deck identification field in the skeleton record is optional. A description of the skeleton record format is contained in appendix P.

The re-entrant FORTRAN runtime library skeleton records are as follows:

```
*LP      REENTRANT FORTRAN RUNTIME LIBRARY
*B  'FORTR'   '  DECK-ID A01  FTN 3.3 RUNTIME'
*B  'Q8PRMR'  '  DECK-ID B01  FTN 3.3 RUNTIME'
*B  'PARABR'  '  DECK-ID B02  FTN 3.3 RUNTIME'
*B  'Q8F2IR'  '  DECK-ID B03  FTN 3.3 RUNTIME'
*B  'ABSR'    '  DECK-ID B04  FTN 3.3 RUNTIME'
*B  'SQRTFR'  '  DECK-ID B05  FTN 3.3 RUNTIME'
*B  'SIGNR'   '  DECK-ID B06  FTN 3.3 RUNTIME'
*B  'FXFLTR'  '  DECK-ID B07  FTN 3.3 RUNTIME'
*B  'EXPR'    '  DECK-ID B08  FTN 3.3 RUNTIME'
*B  'ALOGR'   '  DECK-ID B09  FTN 3.3 RUNTIME'
*B  'TANHR'   '  DECK-ID B10  FTN 3.3 RUNTIME'
*B  'SNCSR'   '  DECK-ID B11  FTN 3.3 RUNTIME'
*B  'ATANR'   '  DECK-ID B12  FTN 3.3 RUNTIME'
*B  'Q8QIOR'  '  DECK-ID C01  FTN 3.3 RUNTIME'
*B  'BINARR'  '  DECK-ID C02  FTN 3.3 RUNTIME'
*B  'IOCODR'  '  DECK-ID D01  FTN 3.3 RUNTIME'
*B  'INITLR'  '  DECK-ID D02  FTN 3.3 RUNTIME'
*B  'RSTORR'  '  DECK-ID D03  FTN 3.3 RUNTIME'
*B  'GETCHR'  '  DECK-ID D04  FTN 3.3 RUNTIME'
*B  'IPACKR'  '  DECK-ID D05  FTN 3.3 RUNTIME'
*B  'UPDATR'  '  DECK-ID D06  FTN 3.3 RUNTIME'
*B  'DECPLR'  '  DECK-ID D07  FTN 3.3 RUNTIME'
*B  'INTGRR'  '  DECK-ID D08  FTN 3.3 RUNTIME'
*B  'SPACER'  '  DECK-ID D09  FTN 3.3 RUNTIME'
*B  'HOLR'    '  DECK-ID D10  FTN 3.3 RUNTIME'
*B  'DCHXR'   '  DECK-ID D11  FTN 3.3 RUNTIME'
*B  'HXASCR'  '  DECK-ID D12  FTN 3.3 RUNTIME'
*B  'AFMTOR'  '  DECK-ID D13  FTN 3.3 RUNTIME'
*B  'RFMTOR'  '  DECK-ID D14  FTN 3.3 RUNTIME'
*B  'AFMTIR'  '  DECK-ID D15  FTN 3.3 RUNTIME'
*B  'RFMTIR'  '  DECK-ID D16  FTN 3.3 RUNTIME'
*B  'ASCHXR'  '  DECK-ID D17  FTN 3.3 RUNTIME'
*B  'HXDCR'   '  DECK-ID D18  FTN 3.3 RUNTIME'
*B  'FLOTIR'  '  DECK-ID D19  FTN 3.3 RUNTIME'
*B  'FOUTR'   '  DECK-ID D20  FTN 3.3 RUNTIME'
*B  'EOUTR'   '  DECK-ID D21  FTN 3.3 RUNTIME'
*B  'EWRITR'  '  DECK-ID D22  FTN 3.3 RUNTIME'
*B  'INTIIR'  '  DECK-ID D23  FTN 3.3 RUNTIME'
*B  'FORMTR'  '  DECK-ID D24  FTN 3.3 RUNTIME'
*B  'Q8QFIR'  '  DECK-ID D25  FTN 3.3 RUNTIME'
*B  'Q8QFLR'  '  DECK-ID D26  FTN 3.3 RUNTIME'
*B  'Q8QFXR'  '  DECK-ID D27  FTN 3.3 RUNTIME'
*B  'HEXAR'   '  DECK-ID D28  FTN 3.3 RUNTIME'
*B  'HEXDR'   '  DECK-ID D29  FTN 3.3 RUNTIME'
*B  'ASCIIR'  '  DECK-ID D30  FTN 3.3 RUNTIME'
*B  'DECHXR'  '  DECK-ID D31  FTN 3.3 RUNTIME'
*B  'AFORMR'  '  DECK-ID D32  FTN 3.3 RUNTIME'
*B  'RFORMR'  '  DECK-ID D33  FTN 3.3 RUNTIME'
*B  'FLOTGR'  '  DECK-ID D34  FTN 3.3 RUNTIME'
*B  'FLOTR'   '  DECK-ID B14  FTN 3.3 RUNTIME'
*B  'COMFPR'  '  DECK-ID B15  FTN 3.3 RUNTIME'
```

The re-entrant FORTRAN runtime library skeleton records include the records if, and only if, double precision FORTRAN is to be in the system.

```
*B  'SQDBLR'  '  DECK-ID E01  FTN 3.3 RUNTIME'
*B  'Q8D2IR'  '  DECK-ID E02  FTN 3.3 RUNTIME'
*B  'DAFSR'   '  DECK-ID E03  FTN 3.3 RUNTIME'
*B  'DSQRTR'  '  DECK-ID E04  FTN 3.3 RUNTIME'
*B  'DSIGNR'  '  DECK-ID E05  FTN 3.3 RUNTIME'
*B  'DEXPR'   '  DECK-ID E08  FTN 3.3 RUNTIME'
*B  'DLOGR'   '  DECK-ID E09  FTN 3.3 RUNTIME'
*B  'DSNCSR'  '  DECK-ID E11  FTN 3.3 RUNTIME'
*B  'DATANR'  '  DECK-ID E12  FTN 3.3 RUNTIME'
*B  'Q8QDFR'  '  DECK-ID E14  FTN 3.3 RUNTIME'
*B  'DOUTR'   '  DECK-ID E15  FTN 3.3 RUNTIME'
*B  'DFLOTR'  '  DECK-ID B13  FTN 3.3 RUNTIME'
*B  'DRSTRR'  '  DECK-ID B19  FTN 3.3 RUNTIME'
```

The appropriate FORTRAN compiler (Version 3.3A or B) skeleton records and the FORTRAN nonre-entrant runtime library records must be inserted into the skeleton. These should be inserted somewhere after the *LIBEDT skeleton record and before the first of the two *Z records at the end of the skeleton. Care must be taken to insert these records so that current system modules linked together on mass memory are not interrupted by the insertions.

To avoid such interruptions, the insertion may be made immediately after the *S skeleton records that define request priorities or immediately before the two final *Z skeleton records.

The FORTRAN Version 3.3A compiler skeleton records are listed as follows. When inserting the skeleton records, the value of p is the logical unit of the installation device. For example, the record *K, I6, is inserted for *K, Ip if 6 is the logical unit number of the installation device.

```
*K,Ip
*LIBEDT
*K,Ip
*L,FTN
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*K,P8
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOA'     '  DECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'  '  DECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'DIAG'    '  DECK-ID F04  FORTRAN 3.3A'
*B  'DIAGRG'  '  DECK-ID F65  FORTRAN 3.3A'
*B  'GETC'    '  DECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'OUTENT'  '  DECK-ID A07  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STORE'   '  DECK-ID F11  FORTRAN 3.3A'
*B  'SYMBOL'  '  DECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAA'  '  DECK-ID F17  FORTRAN 3.3A'
*B  'DUMYAA'  '  DECK-ID F18  FORTRAN 3.3A'
*B  'PHASEA'  '  DECK-ID A08  FORTRAN 3.3A'
*B  'ARAYSZ'  '  DECK-ID A42  FORTRAN 3.3A'
*B  'CPLCCP'  '  DECK-ID A43  FORTRAN 3.3A'
*B  'ENCOO'   '  DECK-ID A29  FORTRAN 3.3A'
*B  'GNST'    '  DECK-ID A06  FORTRAN 3.3A'
*B  'HEADER'  '  DECK-ID F64  FORTRAN 3.3A'
*B  'IGETCF'  '  DECK-ID F14  FORTRAN 3.3A'
*B  'OPTION'  '  DECK-ID F15  FORTRAN 3.3A'
*B  'PLABEL'  '  DECK-ID A09  FORTRAN 3.3A'
*B  'C80BCS'  '  DECK-ID A10  FORTRAN 3.3A'
*B  'RDLABL'  '  DECK-ID A11  FORTRAN 3.3A'
*B  'SAVEID'  '  DECK-ID A04  FORTRAN 3.3A'
*B  'STCHAR'  '  DECK-ID A12  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AA,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOA'     '  DECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'  '  DECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'DIAG'    '  DECK-ID F04  FORTRAN 3.3A'
*B  'DIAGRG'  '  DECK-ID F65  FORTRAN 3.3A'
*B  'GETC'    '  DECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'OUTENT'  '  DECK-ID A07  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STORE'   '  DECK-ID F11  FORTRAN 3.3A'
*B  'SYMBOL'  '  DECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAB'  '  DECK-ID F19  FORTRAN 3.3A'
*B  'DUMYAB'  '  DECK-ID F20  FORTRAN 3.3A'
*B  'BYEOPR'  '  DECK-ID A19  FORTRAN 3.3A'
*B  'CFLOT'   '  DECK-ID F06  FORTRAN 3.3A'
*B  'DUMVOL'  '  DECK-ID F07  FORTRAN 3.3A'
*B  'EXP9'    '  DECK-ID F05  FORTRAN 3.3A'
*B  'GETF'    '  DECK-ID A05  FORTRAN 3.3A'
*B  'GPUT'    '  DECK-ID A02  FORTRAN 3.3A'
*B  'SAVEID'  '  DECK-ID A04  FORTRAN 3.3A'
*B  'STCHAR'  '  DECK-ID A12  FORTRAN 3.3A'
*B  'SUBPPR'  '  DECK-ID A23  FORTRAN 3.3A'
*B  'TYPE'    '  DECK-ID A13  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,Ip
*N,FTN3AB,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOA'     '  DECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'  '  DECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'DIAG'    '  DECK-ID F04  FORTRAN 3.3A'
*B  'DIAGRG'  '  DECK-ID F65  FORTRAN 3.3A'
*B  'GETC'    '  DECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'OUTENT'  '  DECK-ID A07  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STORE'   '  DECK-ID F11  FORTRAN 3.3A'
*B  'SYMBOL'  '  DECK-ID A03  FORTRAN 3.3A'
*B  'LCCLAC'  '  DECK-ID F21  FORTRAN 3.3A'
*B  'DUMYAC'  '  DECK-ID F22  FORTRAN 3.3A'
*B  'ASGNPR'  '  DECK-ID A32  FORTRAN 3.3A'
*B  'BDCPR'   '  DECK-ID A33  FORTRAN 3.3A'
*B  'CFIVCC'  '  DECK-ID A34  FORTRAN 3.3A'
*B  'CKIVC'   '  DECK-ID A35  FORTRAN 3.3A'
*B  'CKNAME'  '  DECK-ID A36  FORTRAN 3.3A'
*B  'COMNPR'  '  DECK-ID A15  FORTRAN 3.3A'
*B  'CFLOT'   '  DECK-ID F06  FORTRAN 3.3A'
*B  'CIMPR'   '  DECK-ID A16  FORTRAN 3.3A'
*B  'CUMVCL'  '  DECK-ID F07  FORTRAN 3.3A'
*B  'CXP9'    '  DECK-ID F05  FORTRAN 3.3A'
*B  'ERRPR'   '  DECK-ID A38  FORTRAN 3.3A'
*B  'EXRLPR'  '  DECK-ID A24  FORTRAN 3.3A'
*B  'GETF'    '  DECK-ID A05  FORTRAN 3.3A'
*B  'GPUT'    '  DECK-ID A02  FORTRAN 3.3A'
*B  'RDLABL'  '  DECK-ID A11  FORTRAN 3.3A'
*B  'TYPEPR'  '  DECK-ID A18  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AC,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOA'     '  DECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'  '  DECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'DIAG'    '  DECK-ID F04  FORTRAN 3.3A'
*B  'DIAGRG'  '  DECK-ID F65  FORTRAN 3.3A'
*B  'GETC'    '  DECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'OUTENT'  '  DECK-ID A07  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STORE'   '  DECK-ID F11  FORTRAN 3.3A'
*B  'SYMBOL'  '  DECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAD'  '  DECK-ID F23  FORTRAN 3.3A'
*B  'DUMYAD'  '  DECK-ID F24  FORTRAN 3.3A'
*B  'ASEMPR'  '  DECK-ID A40  FORTRAN 3.3A'
*B  'CFLOT'   '  DECK-ID F06  FORTRAN 3.3A'
*B  'DUMVOL'  '  DECK-ID F07  FORTRAN 3.3A'
```

```
*B  'CXP9'     '  CECK-ID F05  FORTRAN 3.3A'
*E  'GETF'     '  CECK-ID A05  FORTRAN 3.3A'
*B  'GPUT'     '  CECK-ID A02  FORTRAN 3.3A'
*E  'IGETCF'   '  CECK-ID F14  FORTRAN 3.3A'
*B  'PUNT'     '  CECK-ID A27  FORTRAN 3.3A'
*B  'RDLABL'   '  CECK-ID A11  FORTRAN 3.3A'
*B  'SUBSCR'   '  CECK-ID A17  FORTRAN 3.3A'
*E  'ENDLOC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,Ip
*N,FTN3AD,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GCA'      '  CECK-ID F02  FORTRAN 3.3A'
*B  'ICPRBA'   '  CECK-ID F08  FORTRAN 3.3A'
*E  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'CCNV'     '  CECK-ID F03  FORTRAN 3.3A'
*E  'CIAG'     '  CECK-ID F04  FORTRAN 3.3A'
*B  'CIAGRG'   '  CECK-ID F65  FORTRAN 3.3A'
*E  'GETC'     '  CECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*E  'CUTENT'   '  CECK-ID A07  FORTRAN 3.3A'
*B  'PACK'     '  CECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'   '  CECK-ID F10  FORTRAN 3.3A'
*B  'STORE'    '  CECK-ID F11  FORTRAN 3.3A'
*E  'SYMBOL'   '  CECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAE'   '  CECK-ID F25  FORTRAN 3.3A'
*E  'CUMYAE'   '  CECK-ID F26  FORTRAN 3.3A'
*B  'CCNSUB'   '  CECK-ID A30  FORTRAN 3.3A'
*E  'CATAPR'   '  CECK-ID A31  FORTRAN 3.3A'
*B  'CFLOT'    '  CECK-ID F06  FORTRAN 3.3A'
*B  'CUMVOL'   '  CECK-ID F07  FORTRAN 3.3A'
*B  'CXP9'     '  CECK-ID F05  FORTRAN 3.3A'
*B  'GETF'     '  CECK-ID A05  FORTRAN 3.3A'
*B  'GPUT'     '  CECK-ID A02  FORTRAN 3.3A'
*E  'STCHAR'   '  CECK-ID A12  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AE,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GCA'      '  CECK-ID F02  FORTRAN 3.3A'
*B  'ICPRBA'   '  CECK-ID F08  FORTRAN 3.3A'
*E  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'CONV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'CIAG'     '  CECK-ID F04  FORTRAN 3.3A'
*B  'CIAGRG'   '  CECK-ID F65  FORTRAN 3.3A'
*B  'GETC'     '  CECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*B  'CUTENT'   '  CECK-ID A07  FORTRAN 3.3A'
*B  'PACK'     '  CECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'   '  CECK-ID F10  FORTRAN 3.3A'
*B  'STORE'    '  CECK-ID F11  FORTRAN 3.3A'
*E  'SYMBOL'   '  CECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAF'   '  CECK-ID F27  FORTRAN 3.3A'
*B  'CUMYAF'   '  CECK-ID F28  FORTRAN 3.3A'
*B  'CHECKF'   '  CECK-ID A20  FORTRAN 3.3A'
*E  'FGETC'    '  CECK-ID A21  FORTRAN 3.3A'
*B  'FORK'     '  CECK-ID A22  FORTRAN 3.3A'
*B  'PECVS'    '  CECK-ID A25  FORTRAN 3.3A'
*B  'PRNTNM'   '  CECK-ID A26  FORTRAN 3.3A'
*B  'STCHAR'   '  CECK-ID A12  FORTRAN 3.3A'
*B  'SYMSCN'   '  CECK-ID A28  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AF,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GCA'      '  CECK-ID F02  FORTRAN 3.3A'
*E  'IOPRBA'   '  CECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*E  'CCNV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'CIAG'     '  CECK-ID F04  FORTRAN 3.3A'
*E  'CIAGRG'   '  CECK-ID F65  FORTRAN 3.3A'
*B  'GETC'     '  CECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*E  'CUTENT'   '  CECK-ID A07  FORTRAN 3.3A'
*B  'PACK'     '  CECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'   '  CECK-ID F10  FORTRAN 3.3A'
*B  'STORE'    '  CECK-ID F11  FORTRAN 3.3A'
*E  'SYMBCL'   '  CECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAF'   '  CECK-ID F27  FORTRAN 3.3A'
*B  'CUMYAF'   '  CECK-ID F28  FORTRAN 3.3A'
*B  'CHECKF'   '  CECK-ID A20  FORTRAN 3.3A'
*E  'FGETC'    '  CECK-ID A21  FORTRAN 3.3A'
*B  'FORK'     '  CECK-ID A22  FORTRAN 3.3A'
*B  'PECVS'    '  CECK-ID A25  FORTRAN 3.3A'
*B  'PRNTNM'   '  CECK-ID A26  FORTRAN 3.3A'
*B  'STCHAR'   '  CECK-ID A12  FORTRAN 3.3A'
*B  'SYMSCN'   '  CECK-ID A28  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AG,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GCA'      '  CECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'   '  CECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'CCNV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'CIAG'     '  CECK-ID F04  FORTRAN 3.3A'
*B  'CIAGRG'   '  CECK-ID F65  FORTRAN 3.3A'
*B  'GETC'     '  CECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*B  'CUTENT'   '  CECK-ID A07  FORTRAN 3.3A'
*B  'PACK'     '  CECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'   '  CECK-ID F10  FORTRAN 3.3A'
*B  'STORE'    '  CECK-ID F11  FORTRAN 3.3A'
*B  'SYMBCL'   '  CECK-ID A03  FORTRAN 3.3A'
*E  'LOCLAH'   '  CECK-ID F31  FORTRAN 3.3A'
*B  'CUMYAH'   '  CECK-ID F32  FORTRAN 3.3A'
*B  'IGETCF'   '  CECK-ID F14  FORTRAN 3.3A'
*B  'MCDMXR'   '  CECK-ID A39  FORTRAN 3.3A'
*B  'PUNT'     '  CECK-ID A27  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AH,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  DECK-ID F01  FORTRAN 3.3A'
*B  'GCA'      '  DECK-ID F02  FORTRAN 3.3A'
*B  'IOPRBA'   '  CECK-ID F08  FORTRAN 3.3A'
*B  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'CONV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'CIAG'     '  DECK-ID F04  FORTRAN 3.3A'
*B  'CIAGRG'   '  CECK-ID F65  FORTRAN 3.3A'
```

```
*B  'GETC'     '  CECK-ID F13  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*B  'CUTENT'   '  DECK-ID A07  FORTRAN 3.3A'
*B  'PACK'     '  DECK-ID F09  FORTRAN 3.3A'
*B  'C8PRMS'   '  DECK-ID F10  FORTRAN 3.3A'
*B  'STORE'    '  CECK-ID F11  FORTRAN 3.3A'
*B  'SYMBCL'   '  DECK-ID A03  FORTRAN 3.3A'
*B  'LOCLAI'   '  CECK-ID F33  FORTRAN 3.3A'
*B  'DUMYAI'   '  DECK-ID F34  FORTRAN 3.3A'
*B  'IOSPR'    '  CECK-ID A37  FORTRAN 3.3A'
*B  'ENDDO'    '  CECK-ID A29  FORTRAN 3.3A'
*B  'ROLABL'   '  CECK-ID A11  FORTRAN 3.3A'
*B  'STCHAR'   '  CECK-ID A12  FORTRAN 3.3A'
*B  'ENDLOC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3AI,,,B
*K,Ip
*K,P8
*P
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GOB'      '  CECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'    '  CECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'   '  CECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'   '  DECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'   '  CECK-ID B03  FORTRAN 3.3A'
*B  'KOUTPT'   '  CECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'   '  CECK-ID B05  FORTRAN 3.3A'
*B  'KPC3PR'   '  CECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'   '  CECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'   '  CECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'   '  CECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'     '  CECK-ID B10  FORTRAN 3.3A'
*B  'CONV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'   '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'   '  DECK-ID F37  FORTRAN 3.3A'
*B  'SYMBOL'   '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALCC'   '  CECK-ID B12  FORTRAN 3.3A'
*B  'LOCLBA'   '  DECK-ID F38  FORTRAN 3.3A'
*B  'DUMYBA'   '  DECK-ID F39  FORTRAN 3.3A'
*B  'PHASEB'   '  DECK-ID B21  FORTRAN 3.3A'
*B  'INXRST'   '  CECK-ID B19  FORTRAN 3.3A'
*B  'NOPROC'   '  CECK-ID S20  FORTRAN 3.3A'
*B  'READIR'   '  CECK-ID B22  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3BA,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GOB'      '  DECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'     '  CECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'    '  DECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'   '  CECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'   '  CECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'   '  CECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'   '  DECK-ID B03  FORTRAN 3.3A'
*B  'KOUTPT'   '  CECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'   '  CECK-ID B05  FORTRAN 3.3A'
*B  'KPC3PR'   '  CECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'   '  CECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'   '  CECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'   '  DECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'     '  DECK-ID B10  FORTRAN 3.3A'
*B  'CONV'     '  DECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'   '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'   '  DECK-ID F37  FORTRAN 3.3A'
*B  'SYMBCL'   '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALOC'   '  DECK-ID B12  FORTRAN 3.3A'
*B  'LOCLBB'   '  CECK-ID F40  FORTRAN 3.3A'
*B  'DUMYBB'   '  DECK-ID F41  FORTRAN 3.3A'
*B  'AFIDL'    '  DECK-ID B25  FORTRAN 3.3A'
*B  'ASSEM'    '  DECK-ID B13  FORTRAN 3.3A'
*B  'BANANA'   '  CECK-ID B14  FORTRAN 3.3A'
*B  'END'      '  DECK-ID B16  FORTRAN 3.3A'
*B  'ENTCOD'   '  CECK-ID B17  FORTRAN 3.3A'
*B  'INXRST'   '  DECK-ID B19  FORTRAN 3.3A'
*B  'SUBFUN'   '  CECK-ID B23  FORTRAN 3.3A'
*B  'INTRAM'   '  DECK-ID B29  FORTRAN 3.3A'
*B  'ENDLCC'   '  CECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3BB,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  CECK-ID F01  FORTRAN 3.3A'
*B  'GOB'      '  DECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'     '  DECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'    '  CECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'   '  DECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'   '  DECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'   '  DECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'   '  CECK-ID B03  FORTRAN 3.3A'
*B  'KCUTPT'   '  DECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'   '  CECK-ID B05  FORTRAN 3.3A'
*B  'KPC3PR'   '  DECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'   '  CECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'   '  DECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'   '  CECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'     '  DECK-ID B10  FORTRAN 3.3A'
*B  'CONV'     '  CECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'   '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'   '  CECK-ID F37  FORTRAN 3.3A'
*B  'SYMBCL'   '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALCC'   '  DECK-ID B12  FORTRAN 3.3A'
*B  'LOCLBC'   '  DECK-ID F42  FORTRAN 3.3A'
*B  'DUMYBC'   '  CECK-ID F43  FORTRAN 3.3A'
*B  'ASUPER'   '  DECK-ID B26  FORTRAN 3.3A'
*B  'ARAYSZ'   '  CECK-ID A42  FORTRAN 3.3A'
*B  'BGINDO'   '  CECK-ID B15  FORTRAN 3.3A'
*B  'CGOTO'    '  DECK-ID B27  FORTRAN 3.3A'
*B  'HELEN'    '  CECK-ID B18  FORTRAN 3.3A'
*B  'SYMSCN'   '  CECK-ID A28  FORTRAN 3.3A'
*B  'ENDLCC'   '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3BC,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'   '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOB'      '  DECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'     '  DECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'    '  DECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'   '  CECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'   '  DECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'   '  DECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'   '  DECK-ID B03  FORTRAN 3.3A'
*B  'KOUTPT'   '  CECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'   '  DECK-ID B05  FORTRAN 3.3A'
```

```
*B  'KPC3PR'  '  DECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'  '  DECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'  '  DECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'  '  DECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'    '  DECK-ID B10  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'  '  DECK-ID F37  FORTRAN 3.3A'
*B  'SYMBCL'  '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALCC'  '  DECK-ID B12  FORTRAN 3.3A'
*B  'LOCLBD'  '  DECK-ID F44  FORTRAN 3.3A'
*B  'CUMYBD'  '  DECK-ID F45  FORTRAN 3.3A'
*B  'ARITHR'  '  DECK-ID B34  FORTRAN 3.3A'
*B  'FINK'    '  DECK-ID B28  FORTRAN 3.3A'
*B  'INTRAM'  '  DECK-ID B29  FORTRAN 3.3A'
*B  'ENDLCC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN38D,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOB'     '  DECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'   '  DECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'  '  DECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'  '  DECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'  '  DECK-ID B03  FORTRAN 3.3A'
*B  'KCUTPT'  '  DECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'  '  DECK-ID B05  FORTRAN 3.3A'
*B  'KPC3PR'  '  DECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'  '  DECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'  '  DECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'  '  DECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'    '  DECK-ID B10  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'  '  DECK-ID F37  FORTRAN 3.3A'
*B  'SYMBCL'  '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALCC'  '  DECK-ID B12  FORTRAN 3.3A'
*B  'LCCLBE'  '  DECK-ID F46  FORTRAN 3.3A'
*B  'CUMYBE'  '  DECK-ID F47  FORTRAN 3.3A'
*B  'ACP'     '  DECK-ID B24  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN38E,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOB'     '  DECK-ID F35  FORTRAN 3.3A'
*B  'CNVT'    '  DECK-ID A01  FORTRAN 3.3A'
*B  'DUMMY'   '  DECK-ID B01  FORTRAN 3.3A'
*B  'FCMSTK'  '  DECK-ID B02  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID F12  FORTRAN 3.3A'
*B  'IOPRBB'  '  DECK-ID F36  FORTRAN 3.3A'
*B  'KCPART'  '  DECK-ID B03  FORTRAN 3.3A'
*B  'KCUTPT'  '  DECK-ID B04  FORTRAN 3.3A'
*B  'KPCSTK'  '  DECK-ID B05  FORTRAN 3.3A'
*B  'KPC3PR'  '  DECK-ID B06  FORTRAN 3.3A'
*B  'KSYMGN'  '  DECK-ID B07  FORTRAN 3.3A'
*B  'LABKPC'  '  DECK-ID B08  FORTRAN 3.3A'
*B  'LABLER'  '  DECK-ID B09  FORTRAN 3.3A'
*B  'PUNT'    '  DECK-ID B10  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'STOREB'  '  DECK-ID F37  FORTRAN 3.3A'
*B  'SYMBCL'  '  DECK-ID B11  FORTRAN 3.3A'
*B  'TSALOC'  '  DECK-ID B12  FORTRAN 3.3A'
*B  'LOCLBF'  '  DECK-ID F48  FORTRAN 3.3A'
*B  'CUMYBF'  '  DECK-ID F49  FORTRAN 3.3A'
*B  'SUBPR3'  '  DECK-ID B33  FORTRAN 3.3A'
*B  'INTRAM'  '  DECK-ID B29  FORTRAN 3.3A'
*B  'PARTSB'  '  DECK-ID B30  FORTRAN 3.3A'
*B  'SUBPR1'  '  DECK-ID B31  FORTRAN 3.3A'
*B  'SUBPR2'  '  DECK-ID B32  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN38F,,,B
*K,Ip
*K,P8
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOC'     '  DECK-ID F50  FORTRAN 3.3A'
*B  'ICPRBC'  '  DECK-ID F51  FORTRAN 3.3A'
*B  'BKDWN'   '  DECK-ID C01  FORTRAN 3.3A'
*B  'BLDUP'   '  DECK-ID C02  FORTRAN 3.3A'
*B  'BSS'     '  DECK-ID C03  FORTRAN 3.3A'
*B  'CHKWD'   '  DECK-ID C04  FORTRAN 3.3A'
*B  'CON'     '  DECK-ID C07  FORTRAN 3.3A'
*B  'COUNT'   '  DECK-ID C08  FORTRAN 3.3A'
*B  'DATAST'  '  DECK-ID C09  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID C10  FORTRAN 3.3A'
*B  'INOUT'   '  DECK-ID C11  FORTRAN 3.3A'
*B  'LABEL'   '  DECK-ID C14  FORTRAN 3.3A'
*B  'LABIN'   '  DECK-ID C15  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'REED'    '  DECK-ID C17  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID C19  FORTRAN 3.3A'
*B  'LOCLCA'  '  DECK-ID F52  FORTRAN 3.3A'
*B  'CUMYCA'  '  DECK-ID F53  FORTRAN 3.3A'
*B  'PHASEC'  '  DECK-ID C13  FORTRAN 3.3A'
*B  'ENDLCC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3CA,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOC'     '  DECK-ID F50  FORTRAN 3.3A'
*B  'IOPRBC'  '  DECK-ID F51  FORTRAN 3.3A'
*B  'BKDWN'   '  DECK-ID C01  FORTRAN 3.3A'
*B  'BLDUP'   '  DECK-ID C02  FORTRAN 3.3A'
*B  'BSS'     '  DECK-ID C03  FORTRAN 3.3A'
*B  'CHKWD'   '  DECK-ID C04  FORTRAN 3.3A'
*B  'CON'     '  DECK-ID C07  FORTRAN 3.3A'
*B  'COUNT'   '  DECK-ID C08  FORTRAN 3.3A'
*B  'DATAST'  '  DECK-ID C09  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID C10  FORTRAN 3.3A'
*B  'INOUT'   '  DECK-ID C11  FORTRAN 3.3A'
*B  'LABEL'   '  DECK-ID C14  FORTRAN 3.3A'
*B  'LABIN'   '  DECK-ID C15  FORTRAN 3.3A'
*B  'C8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'REED'    '  DECK-ID C17  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID C19  FORTRAN 3.3A'
*B  'LCCLCB'  '  DECK-ID F54  FORTRAN 3.3A'
*B  'CHOP'    '  DECK-ID C05  FORTRAN 3.3A'
*B  'CL12'    '  DECK-ID C06  FORTRAN 3.3A'
*B  'SKIP'    '  DECK-ID C18  FORTRAN 3.3A'
*B  'IXOPT'   '  DECK-ID C12  FORTRAN 3.3A'
*B  'CXLD'    '  DECK-ID C16  FORTRAN 3.3A'
*B  'ENDLCC'  '  DECK-ID F16  FORTRAN 3.3A'
```

```
*T
*K,I8
*N,FTN3CB,,,B
*K,Ip
*K,P8
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOOD'    '  DECK-ID F55  FORTRAN 3.3A'
*B  'INDEX'   '  DECK-ID D01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID D02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLDA'  '  DECK-ID F58  FORTRAN 3.3A'
*B  'DUMYDA'  '  DECK-ID F59  FORTRAN 3.3A'
*B  'PHASE6'  '  DECK-ID D03  FORTRAN 3.3A'
*B  'BEGINO'  '  DECK-ID D21  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F57  FORTRAN 3.3A'
*B  'FINISH'  '  DECK-ID D22  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID D16  FORTRAN 3.3A'
*B  'IACON'   '  DECK-ID D17  FORTRAN 3.3A'
*B  'IHCON'   '  DECK-ID D18  FORTRAN 3.3A'
*B  'NWRITE'  '  DECK-ID D19  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D20  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3DA,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOOD'    '  DECK-ID F55  FORTRAN 3.3A'
*B  'INDEX'   '  DECK-ID D01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID D02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLDB'  '  DECK-ID F60  FORTRAN 3.3A'
*B  'DUMYDB'  '  DECK-ID F61  FORTRAN 3.3A'
*B  'AMCUT'   '  DECK-ID D04  FORTRAN 3.3A'
*B  'BKDWN'   '  DECK-ID D06  FORTRAN 3.3A'
*B  'COUNT'   '  DECK-ID D07  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID D14  FORTRAN 3.3A'
*B  'LABOUT'  '  DECK-ID D08  FORTRAN 3.3A'
*B  'NP2OUT'  '  DECK-ID D09  FORTRAN 3.3A'
*B  'RBDX'    '  DECK-ID D10  FORTRAN 3.3A'
*B  'RBPK'    '  DECK-ID D11  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D15  FORTRAN 3.3A'
*B  'TABDEC'  '  DECK-ID D12  FORTRAN 3.3A'
*B  'UNFUNC'  '  DECK-ID D13  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3DB,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOOD'    '  DECK-ID F55  FORTRAN 3.3A'
*B  'INDEX'   '  DECK-ID D01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID D02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLDC'  '  DECK-ID F62  FORTRAN 3.3A'
*B  'ADMAX'   '  DECK-ID D05  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID D14  FORTRAN 3.3A'
*B  'TABDEC'  '  DECK-ID D12  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D15  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3DC,,,B
*K,Ip
*K,P8
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOE'     '  DECK-ID F63  FORTRAN 3.3A'
*B  'INDEX'   '  DECK-ID E01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID E02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLCA'  '  DECK-ID F58  FORTRAN 3.3A'
*B  'DUMYCA'  '  DECK-ID F59  FORTRAN 3.3A'
*B  'PHASE6'  '  DECK-ID E03  FORTRAN 3.3A'
*B  'BEGINO'  '  DECK-ID E19  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F57  FORTRAN 3.3A'
*B  'FINISH'  '  DECK-ID E20  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID E14  FORTRAN 3.3A'
*B  'IACON'   '  DECK-ID E15  FORTRAN 3.3A'
*B  'IHCON'   '  DECK-ID E16  FORTRAN 3.3A'
*B  'NWRITE'  '  DECK-ID E17  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'SETPRT'  '  DECK-ID E18  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D20  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3EA,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOE'     '  DECK-ID F63  FORTRAN 3.3A'
*B  'INDEX'   '  DECK-ID E01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID E02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLDB'  '  DECK-ID F60  FORTRAN 3.3A'
*B  'DUMYDB'  '  DECK-ID F61  FORTRAN 3.3A'
*B  'AMCUT'   '  DECK-ID E04  FORTRAN 3.3A'
*B  'BKDWN'   '  DECK-ID E06  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F57  FORTRAN 3.3A'
*B  'COUNT'   '  DECK-ID E07  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID E14  FORTRAN 3.3A'
*B  'IACON'   '  DECK-ID E15  FORTRAN 3.3A'
*B  'IHCON'   '  DECK-ID E16  FORTRAN 3.3A'
*B  'LABOUT'  '  DECK-ID E08  FORTRAN 3.3A'
*B  'NP2OUT'  '  DECK-ID E09  FORTRAN 3.3A'
*B  'NWRITE'  '  DECK-ID E17  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'RBDX'    '  DECK-ID E10  FORTRAN 3.3A'
*B  'RBPK'    '  DECK-ID E11  FORTRAN 3.3A'
*B  'SETPRT'  '  DECK-ID E18  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D20  FORTRAN 3.3A'
*B  'TABDEC'  '  DECK-ID E12  FORTRAN 3.3A'
*B  'UNFUNC'  '  DECK-ID E13  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3EB,,,B
*K,I6
*K,F8
```

```
*P,,,MARKER
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOE'     '  DECK-ID F63  FORTRAN 3.3A'
*B  'INOEX'   '  DECK-ID E01  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'NPUNCH'  '  DECK-ID E02  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'LOCLOC'  '  DECK-ID F62  FORTRAN 3.3A'
*B  'ADMAX'   '  DECK-ID E05  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID E14  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D20  FORTRAN 3.3A'
*B  'TABDEC'  '  DECK-ID E12  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3EC,,,B
*K,Ip
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'GOF'     '  DECK-ID F66  FORTRAN 3.3A'
*B  'PHASEF'  '  DECK-ID G01  FORTRAN 3.3A'
*B  'GETSYM'  '  DECK-ID G02  FORTRAN 3.3A'
*B  'ACON'    '  DECK-ID G03  FORTRAN 3.3A'
*B  'HCON'    '  DECK-ID G04  FORTRAN 3.3A'
*B  'LWRITE'  '  DECK-ID G05  FORTRAN 3.3A'
*B  'MATCH'   '  DECK-ID G06  FORTRAN 3.3A'
*B  'SORT'    '  DECK-ID G07  FORTRAN 3.3A'
*B  'IREPAK'  '  DECK-ID G08  FORTRAN 3.3A'
*B  'SYMSCN'  '  DECK-ID D20  FORTRAN 3.3A'
*B  'CONV'    '  DECK-ID F03  FORTRAN 3.3A'
*B  'PACK'    '  DECK-ID F09  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'Q8PRMS'  '  DECK-ID F10  FORTRAN 3.3A'
*B  'GETSYR'  '  DECK-ID F67  FORTRAN 3.3A'
*B  'TITLE'   '  DECK-ID F68  FORTRAN 3.3A'
*B  'IFCVPF'  '  DECK-ID F69  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3FA,,,B
*K,Ip
*P
*B  'FTN33A'  '  DECK-ID F01  FORTRAN 3.3A'
*B  'ERRMSG'  '  DECK-ID F70  FORTRAN 3.3A'
*B  'IOPRBD'  '  DECK-ID F56  FORTRAN 3.3A'
*B  'ENDLOC'  '  DECK-ID F16  FORTRAN 3.3A'
*T
*K,I8
*N,FTN3ER,,,B
```

The FORTRAN Version 3.3B compiler skeleton records are listed as follows. When inserting the skeleton records, the value of p is the logical unit of the installation device; thus *K, I6 is substituted for *K, Ip if the installation device is logical unit 6.

```
*JOB,INSTAL,FTN 3.3B COMPILER
*K,I6
*LIBEDT
*K,Ip
*L,FTN
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*K,P8
*P
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOA'     '  DECK-ID 02F  FORTRAN 3.3B'
*B  'PHASEA'  '  DECK-ID 07A  FORTRAN 3.3B'
*B  'IOPRBA'  '  DECK-ID 08F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CFIVOC'  '  DECK-ID 34A  FORTRAN 3.3B'
*B  'CKNAME'  '  DECK-ID 36A  FORTRAN 3.3B'
*B  'CNVT'    '  DECK-ID 01A  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'DIAG'    '  DECK-ID 04F  FORTRAN 3.3B'
*B  'DIAGRG'  '  DECK-ID 37F  FORTRAN 3.3B'
*B  'DXP9'    '  DECK-ID 05F  FORTRAN 3.3B'
*B  'DFLOT'   '  DECK-ID 06F  FORTRAN 3.3B'
*B  'DUMVOL'  '  DECK-ID 35F  FORTRAN 3.3B'
*B  'GETC'    '  DECK-ID 14F  FORTRAN 3.3B'
*B  'GETF'    '  DECK-ID 04A  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 07F  FORTRAN 3.3B'
*B  'GPUT'    '  DECK-ID 02A  FORTRAN 3.3B'
*B  'IGETCF'  '  DECK-ID 15F  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RDLABL'  '  DECK-ID 10A  FORTRAN 3.3B'
*B  'STORE'   '  DECK-ID 11F  FORTRAN 3.3B'
*B  'SYMBOL'  '  DECK-ID 03A  FORTRAN 3.3B'
*B  'ENDDO'   '  DECK-ID 29A  FORTRAN 3.3B'
*B  'GNST'    '  DECK-ID 05A  FORTRAN 3.3B'
*B  'HEADER'  '  DECK-ID 36F  FORTRAN 3.3B'
*B  'OPTION'  '  DECK-ID 16F  FORTRAN 3.3B'
*B  'OUTENT'  '  DECK-ID 06A  FORTRAN 3.3B'
*B  'PLABEL'  '  DECK-ID 08A  FORTRAN 3.3B'
*B  'STCHAR'  '  DECK-ID 11A  FORTRAN 3.3B'
*B  'TYPE'    '  DECK-ID 12A  FORTRAN 3.3B'
*B  'SAVEID'  '  DECK-ID 13A  FORTRAN 3.3B'
*B  'LOCLA1'  '  DECK-ID 12F  FORTRAN 3.3B'
*B  'DUMYA1'  '  DECK-ID 13F  FORTRAN 3.3B'
*B  'Q8QBDS'  '  DECK-ID 09A  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3A1,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOA'     '  DECK-ID 02F  FORTRAN 3.3B'
*B  'PHASEA'  '  DECK-ID 07A  FORTRAN 3.3B'
*B  'IOPRBA'  '  DECK-ID 08F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CFIVOC'  '  DECK-ID 34A  FORTRAN 3.3B'
*B  'CKNAME'  '  DECK-ID 36A  FORTRAN 3.3B'
*B  'CNVT'    '  DECK-ID 01A  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'DIAG'    '  DECK-ID 04F  FORTRAN 3.3B'
*B  'DIAGRG'  '  DECK-ID 37F  FORTRAN 3.3B'
*B  'DXP9'    '  DECK-ID 05F  FORTRAN 3.3B'
*B  'DFLOT'   '  DECK-ID 06F  FORTRAN 3.3B'
*B  'DUMVOL'  '  DECK-ID 35F  FORTRAN 3.3B'
*B  'GETC'    '  DECK-ID 14F  FORTRAN 3.3B'
*B  'GETF'    '  DECK-ID 04A  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 07F  FORTRAN 3.3B'
*B  'GPUT'    '  DECK-ID 02A  FORTRAN 3.3B'
*B  'IGETCF'  '  DECK-ID 15F  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RDLABL'  '  DECK-ID 10A  FORTRAN 3.3B'
*B  'STORE'   '  DECK-ID 11F  FORTRAN 3.3B'
*B  'SYMBOL'  '  DECK-ID 03A  FORTRAN 3.3B'
*B  'ENDDO'   '  DECK-ID 29A  FORTRAN 3.3B'
*B  'GNST'    '  DECK-ID 05A  FORTRAN 3.3B'
*B  'HEADER'  '  DECK-ID 36F  FORTRAN 3.3B'
*B  'OPTION'  '  DECK-ID 16F  FORTRAN 3.3B'
*B  'OUTENT'  '  DECK-ID 06A  FORTRAN 3.3B'
*B  'PLABEL'  '  DECK-ID 08A  FORTRAN 3.3B'
*B  'STCHAR'  '  DECK-ID 11A  FORTRAN 3.3B'
*B  'TYPE'    '  DECK-ID 12A  FORTRAN 3.3B'
```

```
*B  'SAVEID'  '  DECK-ID 13A  FORTRAN 3.3B'
*B  'LOCLA2'  '  DECK-ID 18F  FORTRAN 3.3B'
*B  'DUMYA2'  '  DECK-ID 19F  FORTRAN 3.3B'
*B  'BYEOPR'  '  DECK-ID 19A  FORTRAN 3.3B'
*B  'CHECKF'  '  DECK-ID 20A  FORTRAN 3.3B'
*B  'COMNPR'  '  DECK-ID 15A  FORTRAN 3.3B'
*B  'CONSUB'  '  DECK-ID 30A  FORTRAN 3.3B'
*B  'DATAPR'  '  DECK-ID 31A  FORTRAN 3.3B'
*B  'DIMPR'   '  DECK-ID 16A  FORTRAN 3.3B'
*B  'EXRLPR'  '  DECK-ID 24A  FORTRAN 3.3B'
*B  'FGETC'   '  DECK-ID 21A  FORTRAN 3.3B'
*B  'FORK'    '  DECK-ID 22A  FORTRAN 3.3B'
*B  'SUBPPR'  '  DECK-ID 23A  FORTRAN 3.3B'
*B  'TYPEPR'  '  DECK-ID 18A  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3A2,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOA'     '  DECK-ID 02F  FORTRAN 3.3B'
*B  'PHASEA'  '  DECK-ID 07A  FORTRAN 3.3B'
*B  'IOPRBA'  '  DECK-ID 08F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CFIVOC'  '  DECK-ID 34A  FORTRAN 3.3B'
*B  'CKNAME'  '  DECK-ID 36A  FORTRAN 3.3B'
*B  'CNVT'    '  DECK-ID 01A  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'DIAG'    '  DECK-ID 04F  FORTRAN 3.3B'
*B  'DIAGRG'  '  DECK-ID 37F  FORTRAN 3.3B'
*B  'DXP9'    '  DECK-ID 05F  FORTRAN 3.3B'
*B  'DFLOT'   '  DECK-ID 06F  FORTRAN 3.3B'
*B  'DUMVOL'  '  DECK-ID 35F  FORTRAN 3.3B'
*B  'GETC'    '  DECK-ID 14F  FORTRAN 3.3B'
*B  'GETF'    '  DECK-ID 04A  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 07F  FORTRAN 3.3B'
*B  'GPUT'    '  DECK-ID 02A  FORTRAN 3.3B'
*B  'IGETCF'  '  DECK-ID 15F  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RDLABL'  '  DECK-ID 10A  FORTRAN 3.3B'
*B  'STORE'   '  DECK-ID 11F  FORTRAN 3.3B'
*B  'SYMBOL'  '  DECK-ID 03A  FORTRAN 3.3B'
*B  'ENDDO'   '  DECK-ID 29A  FORTRAN 3.3B'
*B  'GNST'    '  DECK-ID 05A  FORTRAN 3.3B'
*B  'HEADER'  '  DECK-ID 36F  FORTRAN 3.3B'
*B  'OPTION'  '  DECK-ID 16F  FORTRAN 3.3B'
*B  'OUTENT'  '  DECK-ID 06A  FORTRAN 3.3B'
*B  'PLABEL'  '  DECK-ID 08A  FORTRAN 3.3B'
*B  'STCHAR'  '  DECK-ID 11A  FORTRAN 3.3B'
*B  'TYPE'    '  DECK-ID 12A  FORTRAN 3.3B'
*B  'SAVEID'  '  DECK-ID 13A  FORTRAN 3.3B'
*B  'LOCLA3'  '  DECK-ID 20F  FORTRAN 3.3B'
*B  'DUMYA3'  '  DECK-ID 21F  FORTRAN 3.3B'
*B  'ARAYSZ'  '  DECK-ID 42A  FORTRAN 3.3B'
*B  'ASEMPR'  '  DECK-ID 40A  FORTRAN 3.3B'
*B  'ASGNPR'  '  DECK-ID 32A  FORTRAN 3.3B'
*B  'BDOPR'   '  DECK-ID 33A  FORTRAN 3.3B'
*B  'CHECKF'  '  DECK-ID 20A  FORTRAN 3.3B'
*B  'CKIVC'   '  DECK-ID 35A  FORTRAN 3.3B'
*B  'CONSUB'  '  DECK-ID 30A  FORTRAN 3.3B'
*B  'CPLOOP'  '  DECK-ID 43A  FORTRAN 3.3B'
*B  'FGETC'   '  DECK-ID 21A  FORTRAN 3.3B'
*B  'FORK'    '  DECK-ID 22A  FORTRAN 3.3B'
*B  'ERBPR'   '  DECK-ID 38A  FORTRAN 3.3B'
*B  'MODMXR'  '  DECK-ID 39A  FORTRAN 3.3B'
*B  'PUNT'    '  DECK-ID 27A  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3A3,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOA'     '  DECK-ID 02F  FORTRAN 3.3B'
*B  'PHASEA'  '  DECK-ID 07A  FORTRAN 3.3B'
*B  'IOPRBA'  '  DECK-ID 08F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CFIVOC'  '  DECK-ID 34A  FORTRAN 3.3B'
*B  'CKNAME'  '  DECK-ID 36A  FORTRAN 3.3B'
*B  'CNVT'    '  DECK-ID 01A  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'DIAG'    '  DECK-ID 04F  FORTRAN 3.3B'
*B  'DIAGRG'  '  DECK-ID 37F  FORTRAN 3.3B'
*B  'DXP9'    '  DECK-ID 05F  FORTRAN 3.3B'
*B  'DFLOT'   '  DECK-ID 06F  FORTRAN 3.3B'
*B  'DUMVOL'  '  DECK-ID 35F  FORTRAN 3.3B'
*B  'GETC'    '  DECK-ID 14F  FORTRAN 3.3B'
*B  'GETF'    '  DECK-ID 04A  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 07F  FORTRAN 3.3B'
*B  'GPUT'    '  DECK-ID 02A  FORTRAN 3.3B'
*B  'IGETCF'  '  DECK-ID 15F  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RDLABL'  '  DECK-ID 10A  FORTRAN 3.3B'
*B  'STORE'   '  DECK-ID 11F  FORTRAN 3.3B'
*B  'SYMBOL'  '  DECK-ID 03A  FORTRAN 3.3B'
*B  'ENDDO'   '  DECK-ID 29A  FORTRAN 3.3B'
*B  'GNST'    '  DECK-ID 05A  FORTRAN 3.3B'
*B  'HEADER'  '  DECK-ID 36F  FORTRAN 3.3B'
*B  'OPTION'  '  DECK-ID 16F  FORTRAN 3.3B'
*B  'OUTENT'  '  DECK-ID 06A  FORTRAN 3.3B'
*B  'PLABEL'  '  DECK-ID 08A  FORTRAN 3.3B'
*B  'STCHAR'  '  DECK-ID 11A  FORTRAN 3.3B'
*B  'TYPE'    '  DECK-ID 12A  FORTRAN 3.3B'
*B  'SAVEID'  '  DECK-ID 13A  FORTRAN 3.3B'
*B  'LOCLA4'  '  DECK-ID 22F  FORTRAN 3.3B'
*B  'DUMYA4'  '  DECK-ID 23F  FORTRAN 3.3B'
*B  'ARITH'   '  DECK-ID 14A  FORTRAN 3.3B'
*B  'SUBSCR'  '  DECK-ID 17A  FORTRAN 3.3B'
*B  'TREE'    '  DECK-ID 41A  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3A4,,,B
*K,Ip
*K,P8
*P,,,MARKER
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOA'     '  DECK-ID 02F  FORTRAN 3.3B'
*B  'PHASEA'  '  DECK-ID 07A  FORTRAN 3.3B'
*B  'IOPRBA'  '  DECK-ID 08F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CFIVOC'  '  DECK-ID 34A  FORTRAN 3.3B'
*B  'CKNAME'  '  DECK-ID 36A  FORTRAN 3.3B'
*B  'CNVT'    '  DECK-ID 01A  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'DIAG'    '  DECK-ID 04F  FORTRAN 3.3B'
*B  'DIAGRG'  '  DECK-ID 37F  FORTRAN 3.3B'
*B  'DXP9'    '  DECK-ID 05F  FORTRAN 3.3B'
*B  'DFLOT'   '  DECK-ID 06F  FORTRAN 3.3B'
*B  'DUMVOL'  '  DECK-ID 35F  FORTRAN 3.3B'
*B  'GETC'    '  DECK-ID 14F  FORTRAN 3.3B'
*B  'GETF'    '  DECK-ID 04A  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 07F  FORTRAN 3.3B'
*B  'GPUT'    '  DECK-ID 02A  FORTRAN 3.3B'
*B  'IGETCF'  '  DECK-ID 15F  FORTRAN 3.3B'
```

```
*B  'PACK'     '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RDLABL'   '  DECK-ID 10A  FORTRAN 3.3B'
*B  'STORE'    '  DECK-ID 11F  FORTRAN 3.3B'
*B  'SYMBOL'   '  DECK-ID 03A  FORTRAN 3.3B'
*B  'ENDDO'    '  DECK-ID 29A  FORTRAN 3.3B'
*B  'GNST'     '  DECK-ID 05A  FORTRAN 3.3B'
*B  'HEADER'   '  DECK-ID 36F  FORTRAN 3.3B'
*B  'OPTION'   '  DECK-ID 16F  FORTRAN 3.3B'
*B  'OUTENT'   '  DECK-ID 06A  FORTRAN 3.3B'
*B  'PLABEL'   '  DECK-ID 08A  FORTRAN 3.3B'
*B  'STCHAR'   '  DECK-ID 11A  FORTRAN 3.3B'
*B  'TYPE'     '  DECK-ID 12A  FORTRAN 3.3B'
*B  'SAVEID'   '  DECK-ID 13A  FORTRAN 3.3B'
*B  'LOCLA5'   '  DECK-ID 24F  FORTRAN 3.3B'
*B  'DUMYA5'   '  DECK-ID 25F  FORTRAN 3.3B'
*B  'BDOPR'    '  DECK-ID 33A  FORTRAN 3.3B'
*B  'CKIVC'    '  DECK-ID 35A  FORTRAN 3.3B'
*B  'IOSPR'    '  DECK-ID 37A  FORTRAN 3.3B'
*B  'PEQVS'    '  DECK-ID 25A  FORTRAN 3.3B'
*B  'PRNTNM'   '  DECK-ID 26A  FORTRAN 3.3B'
*B  'SYMSCN'   '  DECK-ID 28A  FORTRAN 3.3B'
*B  'ENDLOC'   '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3A5,,,B
*K,Ip
*K,P8
*P
*B  'FTN33B'   '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOB'      '  DECK-ID 26F  FORTRAN 3.3B'
*B  'PHASEB'   '  DECK-ID 21B  FORTRAN 3.3B'
*B  'IOPR8B'   '  DECK-ID 27F  FORTRAN 3.3B'
*B  'Q8PRMS'   '  DECK-ID 10F  FORTRAN 3.3B'
*B  'CNVT'     '  DECK-ID 01A  FORTRAN 3.3B'
*B  'DUMMY'    '  DECK-ID 01B  FORTRAN 3.3B'
*B  'FCMSTK'   '  DECK-ID 02B  FORTRAN 3.3B'
*B  'GETSYM'   '  DECK-ID 07F  FORTRAN 3.3B'
*B  'KCPART'   '  DECK-ID 03B  FORTRAN 3.3B'
*B  'KOUTPT'   '  DECK-ID 04B  FORTRAN 3.3B'
*B  'KPCSTK'   '  DECK-ID 05B  FORTRAN 3.3B'
*B  'KPC3PR'   '  DECK-ID 06B  FORTRAN 3.3B'
*B  'KSYMGN'   '  DECK-ID 07B  FORTRAN 3.3B'
*B  'LABKPC'   '  DECK-ID 08B  FORTRAN 3.3B'
*B  'LABLER'   '  DECK-ID 09B  FORTRAN 3.3B'
*B  'PUNT'     '  DECK-ID 10B  FORTRAN 3.3B'
*B  'CONV'     '  DECK-ID 03F  FORTRAN 3.3B'
*B  'STOREB'   '  DECK-ID 34F  FORTRAN 3.3B'
*B  'SYMBOL'   '  DECK-ID 11B  FORTRAN 3.3B'
*B  'TSALOC'   '  DECK-ID 12B  FORTRAN 3.3B'
*B  'ARAYSZ'   '  DECK-ID 42A  FORTRAN 3.3B'
*B  'ASSEM'    '  DECK-ID 13B  FORTRAN 3.3B'
*B  'BANANA'   '  DECK-ID 14B  FORTRAN 3.3B'
*B  'BGINDO'   '  DECK-ID 15B  FORTRAN 3.3B'
*B  'END'      '  DECK-ID 16B  FORTRAN 3.3B'
*B  'ENTCOD'   '  DECK-ID 17B  FORTRAN 3.3B'
*B  'HELEN'    '  DECK-ID 18B  FORTRAN 3.3B'
*B  'INXRST'   '  DECK-ID 19B  FORTRAN 3.3B'
*B  'NOPROC'   '  DECK-ID 20B  FORTRAN 3.3B'
*B  'READIR'   '  DECK-ID 22B  FORTRAN 3.3B'
*B  'SUBFUN'   '  DECK-ID 23B  FORTRAN 3.3B'
*B  'SYMSCN'   '  DECK-ID 29A  FORTRAN 3.3B'
*B  'ACP'      '  DECK-ID 24B  FORTRAN 3.3B'
*B  'AFIDL'    '  DECK-ID 25B  FORTRAN 3.3B'
*B  'ASUPER'   '  DECK-ID 26B  FORTRAN 3.3B'
*B  'CGOTO'    '  DECK-ID 27B  FORTRAN 3.3B'
*B  'FINK'     '  DECK-ID 28B  FORTRAN 3.3B'
*B  'INTRAM'   '  DECK-ID 29B  FORTRAN 3.3B'
*B  'PARTSB'   '  DECK-ID 30B  FORTRAN 3.3B'
*B  'SUBPR1'   '  DECK-ID 31B  FORTRAN 3.3B'
*B  'SUBPR2'   '  DECK-ID 32B  FORTRAN 3.3B'
*B  'SUBPR3'   '  DECK-ID 33B  FORTRAN 3.3B'
*B  'ARITHR'   '  DECK-ID 34B  FORTRAN 3.3B'
*B  'ENDLOC'   '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3B1,,,B
*K,Ip
*K,P8
*P
*B  'FTN33B'   '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOC'      '  DECK-ID 28F  FORTRAN 3.3B'
*B  'PHASEC'   '  DECK-ID 13C  FORTRAN 3.3B'
*B  'IOPR8C'   '  DECK-ID 29F  FORTRAN 3.3B'
*B  'Q8PRMS'   '  DECK-ID 10F  FORTRAN 3.3B'
*B  'BKDWN'    '  DECK-ID 01C  FORTRAN 3.3B'
*B  'BLDUP'    '  DECK-ID 02C  FORTRAN 3.3B'
*B  'BSS'      '  DECK-ID 03C  FORTRAN 3.3B'
*B  'CHKWD'    '  DECK-ID 04C  FORTRAN 3.3B'
*B  'CHOP'     '  DECK-ID 05C  FORTRAN 3.3B'
*B  'CL12'     '  DECK-ID 06C  FORTRAN 3.3B'
*B  'CON'      '  DECK-ID 07C  FORTRAN 3.3B'
*B  'COUNT'    '  DECK-ID 08C  FORTRAN 3.3B'
*B  'DATAST'   '  DECK-ID 09C  FORTRAN 3.3B'
*B  'GETSYM'   '  DECK-ID 10C  FORTRAN 3.3B'
*B  'INOUT'    '  DECK-ID 11C  FORTRAN 3.3B'
*B  'IXOPT'    '  DECK-ID 12C  FORTRAN 3.3B'
*B  'LABEL'    '  DECK-ID 14C  FORTRAN 3.3B'
*B  'LABIN'    '  DECK-ID 15C  FORTRAN 3.3B'
*B  'QXLD'     '  DECK-ID 16C  FORTRAN 3.3B'
*B  'REED'     '  DECK-ID 17C  FORTRAN 3.3B'
*B  'SKIP'     '  DECK-ID 18C  FORTRAN 3.3B'
*B  'SYMSCN'   '  DECK-ID 19C  FORTRAN 3.3B'
*B  'ENDLOC'   '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3C1,,,B
*K,Ip
*K,P8
*P
*B  'FTN33B'   '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOOD'     '  DECK-ID 30F  FORTRAN 3.3B'
*B  'PHASE6'   '  DECK-ID 14D  FORTRAN 3.3B'
*B  'IOPR8D'   '  DECK-ID 31F  FORTRAN 3.3B'
*B  'Q8PRMS'   '  DECK-ID 10F  FORTRAN 3.3B'
*B  'AMOUT'    '  DECK-ID 01D  FORTRAN 3.3B'
*B  'ADMAX'    '  DECK-ID 02D  FORTRAN 3.3B'
*B  'BEGINO'   '  DECK-ID 03D  FORTRAN 3.3B'
*B  'BKDWN'    '  DECK-ID 04D  FORTRAN 3.3B'
*B  'COUNT'    '  DECK-ID 05D  FORTRAN 3.3B'
*B  'FINISH'   '  DECK-ID 06D  FORTRAN 3.3B'
*B  'GETSYM'   '  DECK-ID 10C  FORTRAN 3.3B'
*B  'IACON'    '  DECK-ID 07D  FORTRAN 3.3B'
*B  'IHCON'    '  DECK-ID 08D  FORTRAN 3.3B'
*B  'INDEX'    '  DECK-ID 09D  FORTRAN 3.3B'
*B  'LABOUT'   '  DECK-ID 10D  FORTRAN 3.3B'
*B  'NP2OUT'   '  DECK-ID 11D  FORTRAN 3.3B'
*B  'NPUNCH'   '  DECK-ID 12D  FORTRAN 3.3B'
*B  'NWRITE'   '  DECK-ID 13D  FORTRAN 3.3B'
*B  'PACK'     '  DECK-ID 09F  FORTRAN 3.3B'
*B  'R8DX'     '  DECK-ID 15D  FORTRAN 3.3B'
*B  'R8PK'     '  DECK-ID 16D  FORTRAN 3.3B'
*B  'SYMSCN'   '  DECK-ID 17D  FORTRAN 3.3B'
*B  'TABDEC'   '  DECK-ID 18D  FORTRAN 3.3B'
*B  'UNPUNC'   '  DECK-ID 19D  FORTRAN 3.3B'
*B  'CONV'     '  DECK-ID 33F  FORTRAN 3.3B'
*B  'ENDLOC'   '  DECK-ID 17F  FORTRAN 3.3B'
*T
```

```
*K,I8
*N,FTN3D1,,,B
*K,Ip
*K,P8
*P
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOE'     '  DECK-ID 32F  FORTRAN 3.3B'
*B  'PHASE6'  '  DECK-ID 14E  FORTRAN 3.3B'
*B  'IOPRBD'  '  DECK-ID 31F  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'AMOUT'   '  DECK-ID 01E  FORTRAN 3.3B'
*B  'ADMAX'   '  DECK-ID 02E  FORTRAN 3.3B'
*B  'BEGINO'  '  DECK-ID 03E  FORTRAN 3.3B'
*B  'BKDWN'   '  DECK-ID 04E  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 33F  FORTRAN 3.3B'
*B  'COUNT'   '  DECK-ID 05E  FORTRAN 3.3B'
*B  'FINISH'  '  DECK-ID 06E  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 10C  FORTRAN 3.3B'
*B  'IACON'   '  DECK-ID 07E  FORTRAN 3.3B'
*B  'IHCON'   '  DECK-ID 08E  FORTRAN 3.3B'
*B  'INDEX'   '  DECK-ID 09E  FORTRAN 3.3B'
*B  'LABOUT'  '  DECK-ID 10E  FORTRAN 3.3B'
*B  'NP2OUT'  '  DECK-ID 11E  FORTRAN 3.3B'
*B  'NPUNCH'  '  DECK-ID 12E  FORTRAN 3.3B'
*B  'NWRITE'  '  DECK-ID 13E  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'RBDX'    '  DECK-ID 15E  FORTRAN 3.3B'
*B  'RBPK'    '  DECK-ID 16E  FORTRAN 3.3B'
*B  'SETPRT'  '  DECK-ID 17E  FORTRAN 3.3B'
*B  'SYMSCN'  '  DECK-ID 17D  FORTRAN 3.3B'
*B  'TABDEC'  '  DECK-ID 18E  FORTRAN 3.3B'
*B  'UNPUNC'  '  DECK-ID 19E  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3E1,,,B
*K,Ip
*P
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'GOF'     '  DECK-ID 38F  FORTRAN 3.3B'
*B  'SYMSCN'  '  DECK-ID 28A  FORTRAN 3.3B'
*B  'PHASEF'  '  DECK-ID 01G  FORTRAN 3.3B'
*B  'Q8PRMS'  '  DECK-ID 10F  FORTRAN 3.3B'
*B  'GETSYM'  '  DECK-ID 02G  FORTRAN 3.3B'
*B  'ACON'    '  DECK-ID 03G  FORTRAN 3.3B'
*B  'HCON'    '  DECK-ID 04G  FORTRAN 3.3B'
*B  'LWRITE'  '  DECK-ID 05G  FORTRAN 3.3B'
*B  'MATCH'   '  DECK-ID 06G  FORTRAN 3.3B'
*B  'SORT'    '  DECK-ID 07G  FORTRAN 3.3B'
*B  'IREPAK'  '  DECK-ID 08G  FORTRAN 3.3B'
*B  'CONV'    '  DECK-ID 03F  FORTRAN 3.3B'
*B  'GETSYR'  '  DECK-ID 39F  FORTRAN 3.3B'
*B  'TITLE'   '  DECK-ID 40F  FORTRAN 3.3B'
*B  'IFOVPF'  '  DECK-ID 41F  FORTRAN 3.3B'
*B  'PACK'    '  DECK-ID 09F  FORTRAN 3.3B'
*B  'IOPRBD'  '  DECK-ID 31F  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
*K,I8
*N,FTN3F1,,,B
*K,Ip
*K,P8
*P
*B  'FTN33B'  '  DECK-ID 01F  FORTRAN 3.3B'
*B  'ERRMSG'  '  DECK-ID 42F  FORTRAN 3.3B'
*B  'IOPRBD'  '  DECK-ID 31F  FORTRAN 3.3B'
*B  'ENDLOC'  '  DECK-ID 17F  FORTRAN 3.3B'
*T
```

```
*K,I8
*N,FTN3ER,,,B
```

The FORTRAN nonre-entrant library skeleton records are the same for both the A and B compilers. These records should be inserted after the compiler skeleton records. They are as follows:

```
*K,Ip
*L,READ
*B  'FORTN'   '  DECK-ID F01  FTN 3.3 RUNTIME'
*L,Q8PREP
*B  'Q8PRMS'  '  DECK-ID G01  FTN 3.3 RUNTIME'
*L,Q8QF2I
*B  'Q8QF2I'  '  DECK-ID G02  FTN 3.3 RUNTIME'
*L,ABS
*B  'ABS'     '  DECK-ID G03  FTN 3.3 RUNTIME'
*L,SQRT
*B  'SQRT'    '  DECK-ID G04  FTN 3.3 RUNTIME'
*L,SIGN
*B  'SIGN'    '  DECK-ID G05  FTN 3.3 RUNTIME'
*L,FLOAT
*B  'FIXFLT'  '  DECK-ID G06  FTN 3.3 RUNTIME'
*L,EXP
*B  'EXP'     '  DECK-ID G07  FTN 3.3 RUNTIME'
*L,ALOG
*B  'ALOG'    '  DECK-ID G08  FTN 3.3 RUNTIME'
*L,TANH
*B  'TANH'    '  DECK-ID G09  FTN 3.3 RUNTIME'
*L,SIN
*B  'SINCOS'  '  DECK-ID G10  FTN 3.3 RUNTIME'
*L,ATAN
*B  'ATAN'    '  DECK-ID G11  FTN 3.3 RUNTIME'
*L,PARABS
*B  'PARABN'  '  DECK-ID G12  FTN 3.3 RUNTIME'
*L,Q8IFRM
*B  'Q8IFRM'  '  DECK-ID H01  FTN 3.3 RUNTIME'
*L,Q8FS
*B  'Q8FS'    '  DECK-ID H02  FTN 3.3 RUNTIME'
*L,Q8TRAN
*B  'Q8TRAN'  '  DECK-ID H03  FTN 3.3 RUNTIME'
*L,Q8QINI
*B  'Q8QINI'  '  DECK-ID H04  FTN 3.3 RUNTIME'
*L,Q8QEND
*B  'Q8QEND'  '  DECK-ID H05  FTN 3.3 RUNTIME'
*L,Q8CMP0
*B  'Q8CMP'   '  DECK-ID H06  FTN 3.3 RUNTIME'
*L,Q8RWBU
*B  'Q8RWBU'  '  DECK-ID H07  FTN 3.3 RUNTIME'
*L,Q8EPRM
*B  'Q8ERRM'  '  DECK-ID H08  FTN 3.3 RUNTIME'
*L,Q8DFNF
*B  'Q8DFIO'  '  DECK-ID H09  FTN 3.3 RUNTIME'
*L,Q8QX
*B  'Q8QX'    '  DECK-ID H10  FTN 3.3 RUNTIME'
*L,Q8QUNI
*B  'Q8QUNI'  '  DECK-ID H11  FTN 3.3 RUNTIME'
*L,Q8FGET
*B  'Q8FGET'  '  DECK-ID H12  FTN 3.3 RUNTIME'
*L,Q8MAGT
*B  'Q8MAGT'  '  DECK-ID H13  FTN 3.3 RUNTIME'
*L,EOF
*B  'TAPCON'  '  DECK-ID H14  FTN 3.3 RUNTIME'
*L,IOCK
*B  'IOCK'    '  DECK-ID H15  FTN 3.3 RUNTIME'
*L,Q8PSE
```

```
*B  'PSSTOP'  '  DECK-ID H16  FTN 3.3 RUNTIME'
*L,Q8PAND
*B  'Q8PAND'  '  DECK-ID H17  FTN 3.3 RUNTIME'
*L,Q8EXP1
*B  'Q8EXP1'  '  DECK-ID H18  FTN 3.3 RUNTIME'
*L,Q8EXP9
*B  'Q8EXP9'  '  DECK-ID H19  FTN 3.3 RUNTIME'
*L,SETBFR
*B  'Q8QGTX'  '  DECK-ID H20  FTN 3.3 RUNTIME'
*L,ENCODE
*B  'IOCODE'  '  DECK-ID J01  FTN 3.3 RUNTIME'
*L,COMMON
*B  'PSUEDO'  '  DECK-ID J02  FTN 3.3 RUNTIME'
*L,IGETCH
*B  'IGETCH'  '  DECK-ID J03  FTN 3.3 RUNTIME'
*L,IPACK
*B  'IPACK'   '  DECK-ID J04  FTN 3.3 RUNTIME'
*L,UPDATE
*B  'UPDATN'  '  DECK-ID J05  FTN 3.3 RUNTIME'
*L,DECPL
*B  'DECPL'   '  DECK-ID J06  FTN 3.3 RUNTIME'
*L,INTGR
*B  'INTGR'   '  DECK-ID J07  FTN 3.3 RUNTIME'
*L,SPACEX
*B  'SPACEN'  '  DECK-ID J08  FTN 3.3 RUNTIME'
*L,HOLRTH
*B  'HOLRTH'  '  DECK-ID J09  FTN 3.3 RUNTIME'
*L,DCHX
*B  'DCHX'    '  DECK-ID J10  FTN 3.3 RUNTIME'
*L,HXASC
*B  'HXASC'   '  DECK-ID J11  FTN 3.3 RUNTIME'
*L,AFRMOT
*B  'AFRMOT'  '  DECK-ID J12  FTN 3.3 RUNTIME'
*L,RFRMOT
*B  'RFRMOT'  '  DECK-ID J13  FTN 3.3 RUNTIME'
*L,AFRMIN
*B  'AFRMIN'  '  DECK-ID J14  FTN 3.3 RUNTIME'
*L,RFRMIN
*B  'RFRMIN'  '  DECK-ID J15  FTN 3.3 RUNTIME'
*L,ASCHX
*B  'ASCHX'   '  DECK-ID J16  FTN 3.3 RUNTIME'
*L,HXDC
*B  'HXDC'    '  DECK-ID J17  FTN 3.3 RUNTIME'
*L,FLOTIN
*B  'FLOTIN'  '  DECK-ID J18  FTN 3.3 RUNTIME'
*L,FOUT
*B  'FOUT'    '  DECK-ID J19  FTN 3.3 RUNTIME'
*L,EOUT
*B  'EOUT'    '  DECK-ID J20  FTN 3.3 RUNTIME'
*L,EWRITE
*B  'EWRITE'  '  DECK-ID J21  FTN 3.3 RUNTIME'
*L,INITL1
*B  'INITL1'  '  DECK-ID J22  FTN 3.3 RUNTIME'
*L,FORMTR
*B  'FORMTN'  '  DECK-ID J23  FTN 3.3 RUNTIME'
*L,Q8QFI
*B  'Q8QFI'   '  DECK-ID J24  FTN 3.3 RUNTIME'
*L,Q8QFL
*B  'Q8QFL'   '  DECK-ID J25  FTN 3.3 RUNTIME'
*L,Q8QFX
*B  'Q8QFX'   '  DECK-ID J26  FTN 3.3 RUNTIME'
*L,HEXASC
*B  'HEXASC'  '  DECK-ID J27  FTN 3.3 RUNTIME'
*L,HEXDEC
*B  'HEXDEC'  '  DECK-ID J28  FTN 3.3 RUNTIME'
*L,ASCII
*B  'ASCII'   '  DECK-ID J29  FTN 3.3 RUNTIME'
*L,DECHEX
*B  'DECHEX'  '  DECK-ID J30  FTN 3.3 RUNTIME'
*L,AFORM
*B  'AFORM'   '  DECK-ID J31  FTN 3.3 RUNTIME'
*L,RFORM
*B  'RFORM'   '  DECK-ID J32  FTN 3.3 RUNTIME'
*L,FLOATG
*B  'FLOATG'  '  DECK-ID J33  FTN 3.3 RUNTIME'
*L,HFLOT
*B  'FLOTN'   '  DECK-ID G14  FTN 3.3 RUNTIME'
*L,SPECOP
*B  'COMNFP'  '  DECK-ID G15  FTN 3.3 RUNTIME'
```

The following skeleton records are a part of the FORTRAN nonre-entrant library skeleton records if FORTRAN double-precision is to be in the system.

```
*L,Q8QD2I
*B  'Q8QD2I'  '  DECK-ID K01  FTN 3.3 RUNTIME'
*L,SNGL
*B  'SGLD8L'  '  DECK-ID K03  FTN 3.3 RUNTIME'
*L,DABS
*B  'DABS'    '  DECK-ID K04  FTN 3.3 RUNTIME'
*L,DSQRT
*B  'DSQRT'   '  DECK-ID K05  FTN 3.3 RUNTIME'
*L,DSIGN
*B  'DSIGN'   '  DECK-ID K06  FTN 3.3 RUNTIME'
*L,DEXP
*B  'DEXP'    '  DECK-ID K08  FTN 3.3 RUNTIME'
*L,DLOG
*B  'DLOG'    '  DECK-ID K09  FTN 3.3 RUNTIME'
*L,DSIN
*B  'DSNCSN'  '  DECK-ID K10  FTN 3.3 RUNTIME'
*L,DATAN
*B  'DATAN'   '  DECK-ID K11  FTN 3.3 RUNTIME'
*L,Q8DXP1
*B  'Q8DXP1'  '  DECK-ID K15  FTN 3.3 RUNTIME'
*L,Q8DXP9
*B  'Q8DXP9'  '  DECK-ID K16  FTN 3.3 RUNTIME'
*L,Q8QDFI
*B  'Q8QDFN'  '  DECK-ID K17  FTN 3.3 RUNTIME'
*L,DOUT
*B  'DOUTN'   '  DECK-ID K18  FTN 3.3 RUNTIME'
*L,HDFLOT
*B  'DFLOTN'  '  DECK-ID K12  FTN 3.3 RUNTIME'
*L,DSTOR1
*B  'DRSTOR'  '  DECK-ID K14  FTN 3.3 RUNTIME'
```

If the above double precision records are not added to the system, the FORTRAN dummy link program is necessary. Add the following records:

```
*L,Q8DXP1
*B,  'DBLDMY'     '  DECK-ID K19  FTN 3.3
  RUNTIME'
```

When adding re-entrant FORTRAN, it is necessary to modify the values of BGNMON and END0V4. Whether or not re-entrant FORTRAN is added, it may be necessary to change the value of N4. Refer to appendix L for a diagram showing BGNMON, END0V4, and N4.

The user should first determine the length of the FORTRAN re-entrant library he is using from the following.

Double-Precision Option	With 1781-1 Hardware Floating Point Unit	With Software Floating Point Unit
Without double-precision option	9,000	10,600
With double-precision option	12,500	16,400

Call this length L. Then compute the new values of BGNMON' and ENDOV4' as follows:

$$BGNMON' = BGNMON - \frac{L}{2}$$

$$END0V4' = END0V4 - \frac{L}{2}\dagger$$

To modify the values of BGNMON and END0V4, the skeleton records defining BGNMON and END0V4 must be modified. These records appear near the beginning of the skeleton. The new records have the form:

*S, BGNMON, n_1

*S, END0V4, n_2

where n_1 and n_2 are the new values. These records are found near the beginning of the skeleton.

The current size of unprotected may be ascertained by dumping the contents of the locations $F6_{16}$ and $F7_{16}$ and computing the difference:

$$V = F6_{16} - F7_{16} = \text{number words of unprotected.}$$

Let V' equal the new size of unprotected.

$$\text{Then } V' = V - \frac{L}{2} - 10$$

Where: $\frac{L}{2}$ is the number of words of re-entrant FORTRAN added. (Ten is subtracted due to the increase in the size of SYSDAT.)

The FORTRAN requirements for unprotected memory are shown below. If the value of V' is less than the required size of unprotected, it may be possible to allow more space in unprotected by decreasing the value of N4. The requirements for N4 are discussed in appendix M. To change N4, modify the skeleton record *S, N4, n so that n is the new value of N4. (This record is near the beginning of the skeleton.)

Compiler	Minimum Number Bytes in Unprotected Memory
FORTRAN Version 3.3A	18,600 (= 9300 words)
FORTRAN Version 3.3B	31,900 (=15,950 words)

NOTE

The size of unprotected memory is also affected by the FORTRAN background library requirements of the user. Each FORTRAN job reads into unprotected memory those FORTRAN background library modules required for linkage. This may include modules that are never used by the job but that are required to avoid unpatched externals. A given job may require in excess of 20,000 bytes for FORTRAN background library modules if the double-precision option is used. After system installation, dummy programs may be written to replace unused modules in the FORTRAN background library, if the size of unprotected memory becomes a problem.

6.3 INCORPORATING CHANGES INTO THE SYSTEM

A new installation file must now be created using the new binary version of SYSDAT, the new skeleton, the old installation file, and the FORTRAN binaries. This is accomplished by using the LIBILD utility. Care must be taken to read the new version of SYSDAT before reading

†Only if the monitor follows directly behind unprotected; if the monitor is in the upper bank (65K words) END0V4 need not be changed.

the old installation file. The first SYSDAT read is the version that is incorporated into the new installation file.

Using the new installation file and the system initializer program, the new system may be loaded. Entries on the comment device are as follows:

Entry	Remarks
MI	Operator has manually interrupted the system
*BATCH	Operator requests batch processing
J	Batch processing in control
*JOB	Operator requests job processor
J	Job processor is in control
*SILP	Operator requests system initializer loading program

When the loading program is in control, the comment device displays:

THE INITIALIZER WILL BE MOVED TO LOCATION xxxx AND EXECUTED TURN OFF PROTEC SWITCH AND TYPE CARRIAGE RETURN

If using a 1700 Series computer system, the operator sets the protect switch to the neutral position and presses carriage return.

If using a CYBER 18-20 computer, the operator presses ESCAPE, types J20, and presses carriage return. This clears program protect, signals a carriage return, and reverts to operator mode.

Ready the card reader if the card reader is the installation device.

The operator then proceeds to initializer execution as described in section 3.8. This is followed by library installation as described in section 3.9. If verification of the augmented system is desired, verify the system (in whole or part) using the procedures of section 4.

MSOS is now ready to operate in its augmented form.

A user who does not have file manager in the MSOS originally ordered from Control Data may add this product to his system. First the user must order the installation materials (see the MSOS Version 5 Ordering Bulletin.)

Binary copies of each file manager program are sent to the user. The user must modify SYSDAT and generate a new skeleton. A new installation file is then created using LIBILD. This installation file is used to build the augmented system.

7.1 SYSDAT MODIFICATION

The SYSDAT changes required are in the preset region and the miscellaneous information portions of SYSDAT. Changes affect two sections of the miscellaneous information portion, file manager data and job processor file parameters.

7.1.1 FILE MANAGER DATA

In the file manager data section, the following code must be deleted:

```
*       THESE ENTRIES ALLOW PROPER
        SYSTEM LINKAGE FOR IF THE FILE

*       MANAGER IS NOT SELECTED.
  SPC   2
  ENT   FSLIST
        ENT   ADRFMS, NUMES
        EQU   FSLIST($7FFF)
        EQU   ADRFMS($7FFF), NUMFSD($7FFF)
```

The deleted code is replaced by the code in figure 7-1.

If there is no timer in the system, add:

```
EQU   FDTPER(1)    FILE/DIRECTORY TIMEOUT
                   PERIOD (1/10 SEC.)
EQU   RPTPER(1)    REQUEST PROCESSOR
                   TIMEOUT PERIOD (1/10 SEC.)
```

If there is a timer in the system, add instead:

```
EQU   FDTPER(10)   FILE/DIRECTORY TIMEOUT
                   PERIOD (1/10 SEC.)
EQU   RPTPER(10)   REQUEST PROCESSOR
                   TIMEOUT PERIOD (1/10 SEC.)
```

If the system library unit is one of the following devices:

1867-10/20	Storage Module Drive
1738-853/854	Disk
1733-1-853/854	Cartridge Disk Controller
1752-3/4	Drum

Add the code:

```
          EXT   BEGFMS
ADRFMS    ADC   BEGFMS    BEGINNING OF FILE
                          MANAGER SPACE ON
                          LIB UNIT
```

```
        ENT   FISLU         LOGICAL UNIT OF FIS DIRECTORY AND BLOCKS
        ENT   MAXMMA        MAXIMUM NO. OF MASS MEMORY ATTEMPTS ON ERROR
        ENT   RPTPER        REQUEST PROCESSOR TIMEOUT PERIOD
        ENT   FIDSEC        FIS DIRECTORY, S SECTOR ADDRESS
        ENT   FIBLSA        SECTOR ADDRESS OF LAST FIS BLOCK
        ENT   FDTPER        FILE/DIRECTORY TIMEOUT PERIOD
        ENT   FIBNIX        INDEX TO THE NEXT AVAILABLE LOCATION IN FIBLSA
        ENT   FSLIST        START OF FILE SPACE LIST
        ENT   FSLLTH        FILE SPACE LIST LENGTH
        ENT   FSLEND        END OF FILE SPACE LIST
        ENT   ADRFMS        BEGINNING OF FILE MANAGER SPACE ON LIB UNIT


        EQU   FISLU(LBUNIT)
        EQU   MAXMMA(1)
```

Figure 7-1. Replacement Code in SYSDAT File Manager Section

If the system library unit is a 1739-1 Cartridge Disk or a 1733-2/856-4 Cartridge Disk Controller (i.e., a 1733-2/856 Cartridge Disk Controller with 4.4 million words), add:

```
ADRFMS  ADC   $5BFB   BEGINNING OF FILE
                      MANAGER SPACE ON
                      LIB UNIT
```

If the system library unit is a 1733-2/856-2 Cartridge Disk Controller (i.e., a 1733-2/856 Cartridge Disk Controller with fewer than 4.4 million words), add:

```
ADRFMS  ADC   $20FD   BEGINNING OF FILE
                      MANAGER SPACE ON
                      LIB UNIT
```

All systems add the information in figure 7-2.

All systems add the information in figure 7-3.

7.1.2 FILE SPACE INFORMATION

File manager unit 1:

If there is no file space on any unit except the library unit, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 1, add the information in figure 7-4.

File manager unit 2:

If file manager unit 1 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 2, add the information in figure 7-5. The definitions for p, m, and n are listed in figure 7-4.

```
********          THE FOLLOWING MUST BE IN ORDER          ********
FIDSEC ADC  0           1. FIS DIRECTORY, S SECTOR ADDRESS
FIBLSA ADC  0           2. SECTOR ADDRESS OF THE LAST FIS BLOCK
FIBNIX ADC  0           3. INDEX TO NEXT AVAILABLE LOCATION IN FIBLSA
FSLIST EQU  FSLIST(*)   4. START OF FILE SPACE LIST

********          START OF LOGICAL UNIT ENTRIES           ********

*          L O G I C A L   U N I T   D A T A ,   U N I T   0
*
       ENT NUMFS0
       EQU NUMFS0(n)   NUMBER OF FILE SECTORS - UNIT 0
```

WHERE: n IS THE NUMBER OF SECTORS IN THE FILE SPACE ON THE LIBRARY UNIT

Figure 7-2. Addition to SYSDAT for File Manager

```
LUE0   VFD  X9/LUEL0,X7/LBUNIT  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
       ADC  0                  ADDRESS OF FILE SPACE POOL
       ADC  0                  NUMBER OF AVAILABLE SECTORS
       ADC  NUMFS0             NUMBER OF SECTORS IN THIS FILE SPACE
       NUM  0,1                THREAD OF ONE SECTOR LONG

       NUM  0,2                THREAD OF TWO SECTORS LONG

       NUM  0,3                THREAD OF THREE SECTORS LONG

LUEL0  EQU  LUEL0(*-LUE0)
```

Figure 7-3. Space Information Addition to SYSDAT for File Manager

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   1
*
        ENT BEGLU1
        ENT NUMFS1
        EQU LUNIT1(p)        LOGICAL UNIT OF FILE MANAGER UNIT 1
        EQU BEGLU1(m)        BEGINNING FILE SECTOR  - UNIT 1
        EQU NUMFS1(n)        NUMBER OF FILE SECTORS - UNIT 1

LUE1    VFD  X9/LUFL1,X7/LUNIT1  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU1              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS1              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL1   EQU  LUEL1(*-LUE1)
```

WHERE: p IS THE LOGICAL UNIT OF THE MASS STORAGE DEVICE USED AS FILE MANAGER UNIT 1.
m IS THE SECTOR NUMBER CORRESPONDING TO THE START OF FILE SPACE ON FILE MANAGER UNIT 1.
n IS THE NUMBER OF SECTORS OF FILE SPACE ON FILE MANAGER UNIT 1.

Figure 7-4. Unit 1 Addition to SYSDAT

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   2
*
        ENT BEGLU2
        ENT NUMFS2
        EQU LUNIT2(p)        LOGICAL UNIT OF FILE MANAGER UNIT 2
        EQU BEGLU2(m)        BEGINNING FILE SECTOR  - UNIT 2
        EQU NUMFS2(n)        NUMBER OF FILE SECTORS - UNIT 2

LUE2    VFD  X9/LUFL2,X7/LUNIT2  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU2              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS2              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL2   EQU  LUEL2(*-LUE2)
```

Figure 7-5. Unit 2 Addition to SYSDAT

File manager unit 3:

If file manager unit 2 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 3, add the information in figure 7-6. The definitions for p, m, and n are listed in figure 7-4.

File manager unit 4:

If file manager unit 3 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 4, add the information in figure 7-7. The definitions for p, m, and n are listed in figure 7-4.

```
*          L O G I C A L   U N I T   D A T A .   U N I T   3
*
        ENT BEGLU3
        ENT NUMFS3
        EQU LUNIT3(p)        LOGICAL UNIT OF FILE MANAGER UNIT 3
        EQU BEGLU3(m)        BEGINNING FILE SECTOR  - UNIT 3
        EQU NUMFS3(n)        NUMBER OF FILE SECTORS - UNIT 3

LUE3    VFD  X9/LUEL3,X7/LUNIT3  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU3              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS3              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL3   EQU  LUEL3(*-LUE3)
```

Figure 7-6. Unit 3 Addition to SYSDAT

```
*          L O G I C A L   U N I T   D A T A .   U N I T   4
*
        ENT BEGLU4
        ENT NUMFS4
        EQU LUNIT4(p)        LOGICAL UNIT OF FILE MANAGER UNIT 4
        EQU BEGLU4(m)        BEGINNING FILE SECTOR  - UNIT 4
        EQU NUMFS4(n)        NUMBER OF FILE SECTORS - UNIT 4

LUE4    VFD  X9/LUEL4,X7/LUNIT4  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU4              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS4              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL4   EQU  LUEL4(*-LUE4)
```

Figure 7-7. Unit 4 Addition to SYSDAT

File manager unit 5:

If file manager unit 4 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 5, add the information in figure 7-8. The definitions for p, m, and n are listed in figure 7-4.

File manager unit 6:

If file manager unit 5 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 6, add the information in figure 7-9. The definitions for p, m, and n are listed in figure 7-4.

File manager unit 7:

If file manager unit 6 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 7, add the information in figure 7-10. The definitions for p, m, and n are listed in figure 7-4.

File manager unit 8:

If file manager unit 7 is the last file space unit for the file manager, proceed to section 7.1.3, File Space List Information. If there is file space on file manager unit 8, add the information in figure 7-11. The definitions for p, m, and n are listed in figure 7-4.

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   5
*
        ENT BEGLU5
        ENT NUMFS5
        EQU LUNIT5(p)        LOGICAL UNIT OF FILE MANAGER UNIT 5
        EQU BEGLU5(m)        BEGINNING FILE SECTOR  - UNIT 5
        EQU NUMFS5(n)        NUMBER OF FILE SECTORS - UNIT 5

LUE5    VFD  X9/LUEL5,X7/LUNIT5  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU5              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS5              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL5   EQU  LUEL5(*-LUE5)
```

Figure 7-8. Unit 5 Addition to SYSDAT

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   6
*
        ENT BEGLU6
        ENT NUMFS6
        EQU LUNIT6(p)        LOGICAL UNIT OF FILE MANAGER UNIT 6
        EQU BEGLU6(m)        BEGINNING FILE SECTOR  - UNIT 6
        EQU NUMFS6(n)        NUMBER OF FILE SECTORS - UNIT 6

LUE6    VFD  X9/LUEL6,X7/LUNIT6  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU6              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS6              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD OF TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL6   EQU  LUEL6(*-LUE6)
```

Figure 7-9. Unit 6 Addition to SYSDAT

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   7
*
        ENT BEGLU7
        ENT NUMFS7
        EQU LUNIT7(p)        LOGICAL UNIT OF FILE MANAGER UNIT 7
        EQU BEGLU7(m)        BEGINNING FILE SECTOR  - UNIT 7
        EQU NUMFS7(n)        NUMBER OF FILE SECTORS - UNIT 7

LUE7    VFD  X9/LUEL7,X7/LUNIT7  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
        ADC  BEGLU7              ADDRESS OF FILE SPACE POOL
        ADC  0                   NUMBER OF AVAILABLE SECTORS
        ADC  NUMFS7              NUMBER OF SECTORS IN THIS FILE SPACE
        NUM  0,1                 THREAD OF ONE SECTOR LONG

        NUM  0,2                 THREAD ON TWO SECTORS LONG

        NUM  0,3                 THREAD OF THREE SECTORS LONG

LUEL7   EQU  LUEL7(*-LUE7)
```

Figure 7-10. Unit 7 Addition to SYSDAT

```
*          L O G I C A L   U N I T   D A T A ,   U N I T   8
*
         ENT BEGLU8
         ENT NUMFS8

         EQU LUNIT8(p)        LOGICAL UNIT OF FILE MANAGER UNIT 8

         EQU BEGLU8(m)        BEGINNING FILE SECTOR - UNIT 8

         EQU NUMFS8(n)        NUMBER OF FILE SECTORS - UNIT 8
         SPC 1
LUE8     VFD X9/LUEL8,X7/LUNIT8  LU ENTRY LENGTH(7/15), LOGICAL UNIT(0-6)
         ADC BEGLU8              ADDRESS OF FILE SPACE POOL
         ADC 0                   NUMBER OF AVAILABLE SECTORS
         ADC NUMFS3              NUMBER OF SECTORS IN THIS FILE SPACE
         NUM 0,1                 THREAD OF ONE SECTOR LONG
         NUM 0,2                 THREAD ON TWO SECTORS LONG
         NUM 0,3                 THREAD OF THREE SECTORS LONG
LUEL8    EQU LUEL8(*-LUE8)
```

Figure 7-11. Unit 8 Addition to SYSDAT

7.1.3 FILE SPACE LIST INFORMATION

All systems add the information in figure 7-12.

7.1.4 LINKAGE OF UNSELECTED ENTRY POINTS

Add the following:

```
         EJT
*        FILE MANAGER DATA
*
         SPC  2
*        LINK UNSELECTED ENTRY POINTS
         SPC  2
```

If there is only one file manager file space unit in the system (the library unit), add codes a through h; if there are two file manager units in the system (units 0 and 1), add codes b through h; if there are three file manager units, add c through h; etc. If there are nine file manager file space units in the system, omit codes a through h.

Code Letter	Code
a	`ENT BEGLU1,NUMFS1` `EQU BEGLU1($7FFF),NUMFS1($7FFF)`
b	`ENT BEGLU2,NUMFS2` `EQU BEGLU2($7FFF),NUMFS2($7FFF)`
c	`ENT BEGLU3,NUMFS3` `EQU BEGLU3($7FFF),NUMFS3($7FFF)`
d	`ENT BEGLU4,NUMFS4` `EQU BEGLU4($7FFF),NUMFS4($7FFF)`
e	`ENT BEGLU5,NUMFS5` `EQU BEGLU5($7FFF),NUMFS5($7FFF)`
f	`ENT BEGLU6,NUMFS6` `EQU BEGLU6($7FFF),NUMFS6($7FFF)`

```
         SPC  2
FSLUTH   EQU  FSLUTH(*-FSLIST)    FILE SPACE LIST LENGTH
         SPC  1
FSLEND   NUM  -0                  END OF FILE SPACE LIST
```

Figure 7-12. Limits for File Manager

Code Letter	Code
g	ENT BEGLU7,NUMFS7 EQU BEGLU7($7FFF),NUMFS7($7FFF)
h	ENT BEGLU8,NUMFS8 EQU BEGLU8($7FFF),NUMFS8($7FFF)

In the job processor file parameters section of SYSDAT. delete the following code:

```
ENT   RELFIL          FILE RELEASE
                      PROCESSOR (DUMMY)

EQU   RELFIL($7FFF)
```

7.1.5 PRESET REGION OF SYSDAT

In the preset region of SYSDAT, add the code in figure 7-13.

7.2 SYSTEM SKELETON MODIFICATION

The current skeleton must first be obtained (refer to appendix N for the method). Skeleton modification procedures are also described in appendix N.

Add the following skeleton records to the core resident program part of the skeleton immediately before the core resident drivers:

```
*LP          FILE MANAGER
*B   'FILMGR' ' DECK-ID F01  FILE MANAGER'
*B   'RSPCV4' ' DECK-ID F02  FILE MANAGER'
*B   'SRHFIS' ' DECK-ID F03  FILE MANAGER'
```

Add the following records to that part of the mass resident skeleton processed by the system initializer. These records must precede the *T END OF SYSTEM record.

```
*         MASS RESIDENT FILE MANAGER
*
*M
*B  'DEFFIL'  '  DECK-ID F05  MSOS 4.1'
*B  'FILSPC'  '  DECK-ID F06  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP01,S
*M
*B  'RELFIL'  '  DECK-ID F08  MSOS 4.1'
*B  'RELSPC'  '  DECK-ID F09  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP02,S
*M
*B  'DEFIDX'  '  DECK-ID F10  MSOS 4.1'
*B  'SORTFM'  '  DECK-ID F11  MSOS 4.1'
*B  'FILSPC'  '  DECK-ID F06  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP03,S
*M
*B  'LOKFIL'  '  DECK-ID F12  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP04,S
*M
*B  'UNLFIL'  '  DECK-ID F13  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP05,S
*M
*B  'STOSEQ'  '  DECK-ID F14  MSOS 4.1'
*B  'FILSPC'  '  DECK-ID F06  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP06,S
*M
*B  'STODIR'  '  DECK-ID F15  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP07,S
*M
*B  'STOIDX'  '  DECK-ID F16  MSOS 4.1'
*B  'HASHCD'  '  DECK-ID F17  MSOS 4.1'
*B  'GETKID'  '  DECK-ID F18  MSOS 4.1'
*B  'FILSPC'  '  DECK-ID F06  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP08,S
*M
*B  'RTVSEQ'  '  DECK-ID F19  MSOS 4.1'
*B  'RTNSPC'  '  DECK-ID F20  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP09,S
*M
*B  'RTVDIR'  '  DECK-ID F21  MSOS 4.1'
*B  'RTNSPC'  '  DECK-ID F20  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP10,S
*M
*B  'RTVIDX'  '  DECK-ID F22  MSOS 4.1'
*B  'HASHCD'  '  DECK-ID F17  MSOS 4.1'
*B  'GETKID'  '  DECK-ID F18  MSOS 4.1'
*B  'RTNSPC'  '  DECK-ID F20  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP11,S
*M
*B  'RTVIDO'  '  DECK-ID F23  MSOS 4.1'
*B  'GETKID'  '  DECK-ID F18  MSOS 4.1'
*B  'RTNSPC'  '  DECK-ID F20  MSOS 4.1'
*B  'RPEND'   '  DECK-ID F07  MSOS 4.1'
*S,FMRP12,S
*M
*B  'FMDUMY'  '  DECK-ID F04  MSOS 4.1'
*S,FMREND,S
```

If the system library unit is one of the following devices:

1867-10/20 Storage Module Drive

1738-853/854 Cartridge Disk

1733-1-853/854 Cartridge Disk Controller

1752-3/4 Drum

```
*          F I L E   M A N A G E R   P R E S E T S

          EXT  DEFFIL
          ALF  3•DEFFIL

          ADC  DEFFIL          DEFINE FILE
*
          EXT  RELFIL
          ALF  3•RELFIL
          ADC  RELFIL          RELEASE FILE
*
          EXT  DEFIDX
          ALF  3•DEFIDX
          ADC  DEFIDX          DEFINE INDEXED FILE
*
          EXT  LOKFIL
          ALF  3•LOKFIL
          ADC  LOKFIL          LOCK FILE
*
          EXT  UNLFIL
          ALF  3•UNLFIL
          ADC  UNLFIL          UNLOCK FILE
*
          EXT  STOSEQ
          ALF  3•STOSEQ
          ADC  STOSEQ          STORE SEQUENTIAL RECORD
*
          EXT  STODIR
          ALF  3•STODIR
          ADC  STODIR          STORE DIRECT
*
          EXT  STOIDX
          ALF  3•STOIDX
          ADC  STOIDX          STORE INDEXED RECORD
*
          EXT  RTVSEQ
          ALF  3•RTVSEQ
          ADC  RTVSEQ          RETRIEVE SEQUENTIAL RECORD
*
          EXT  RTVDIR
          ALF  3•RTVDIR
          ADC  RTVDIR          RETRIEVE DIRECT
*
          EXT  RTVIDX
          ALF  3•RTVIDX
          ADC  RTVIDX          RETRIEVE INDEXED RECORD
*
          EXT  RTVIDO
          ALF  3•RTVIDO
          ADC  RTVIDO          RETRIEVE INDEXED-ORDERED RECORD

*         F I L E   M A N A G E R   F L A G   P R E S E T

          EXT  FMPFLG
          ALF  3•FMPFLG
          ADC  FMPFLG
```

Figure 7-13. PRESET Addition for File Manager

add the records:

```
*S, BEGFMS, S    SPECIFY THE SYSTEM FILE
                 SPACE
*M, BEGFMS+q
*M
*B   'FMDUMY' ' DECK-ID  F24  FILE MANAGER'
```

Where: q is the length of the file space on the library unit.

If there are to be job files stored on pseudo tapes, add the following records:

```
*S, JETLV4, S   SPECIFY THE JOB FILE TABLE
                SPACE
*M, JFTLV4+j
```

Where: j is the number of sectors in the job file table in hexadecimal. The value of j must be such that:

$$1 \leq j \leq 100_{16}\dagger$$

If there are to be one or more pseudo tapes in the system, add the following records:

```
*B 'FMDUMY' ' DECK-ID  F24  FILE MANAGER'
*S, SCSPCS, S  SPECIFY THE CONFIGURATOR
               DATA SPACE
*M, SCSPCS+$1000
*B 'FMDUMY' ' DECK-ID  F24  FILE MANAGER'
*S, SCSPCE, S
*S, LBSPCS, S  SPECIFY THE LIBRARY BUILDER
               DATA SPACE
*M, LBSFCS+$2000
*B 'FMDUMY' ' DECK-ID  F24  FILE MANAGER'
*S, LPSPCE, S
```

If the text editor is to be included in the system, add the following:

```
*V   TEXT EDITOR

*K, Ip
*L, EDITOR
*B 'EDITOR' ' DECK-ID  F25  FILE MANAGER'
*K, P8
*P
*B 'EDITFL' ' DECK-ID  F26  FILE MANAGER'
*B 'NXTLOC' ' NEXT AVAILABLE LOCATION'
*T
*K, I8
*N, EDITFL... B
*K, Ip
```

Where: p is the logical unit number of the installation device.

The value of BGNMON must be decreased by 1339 to allow room for the main memory resident file manager modules. If the monitor follows directly behind unprotected, END0V4 must also be changed by the same value; otherwise, END0V4 remains unaltered. The skeleton records defining BGNMON and END0V4 have the form:

```
*S, BGNMON, n1
*S, END0V4, n2
```

These records appear near the beginning of the skeleton. After modification, the values of n_1 and n_2 are the new values of BGNMON and END0V4, respectively.

It may also be necessary to modify the value of N4. A discussion of the requirements for N4 is found in appendix M. The system requirements for unprotected must also be considered before modifying N4.

7.3 INCORPORATING CHANGES INTO SYSTEM

A new installation file must now be created using the new binary version of SYSDAT, the new skeleton, the old installation file, and the file manager binaries. This is accomplished by using LIBILD. Care must be taken so that the new version of SYSDAT is read by LIBILD before reading the old installation file, so that the proper version of SYSDAT is incorporated into the new installation file.

Using the new installation file and the system initializer program, the new system may be loaded. Entries on the comment device are as follows:

Entry	Remarks
MI	Operator has manually interrupted the system
*BATCH	Operator requests batch processing
J	Batch processing is in control
*JOB	Operator requests the job processor
J	Job processor is in control
SILP	Operator requests the system initializer loading program

†There are 9 job files per sector.

The system replies:

```
THE INITIALIZER WILL BE MOVED TO LOCATION
xxxx AND EXECUTED.  TURN OFF PROTEC
SWITCH AND TYPE CARRIAGE RETURN
```

If using a 1700 Series computer system, the operator sets the protect switch to the neutral position and presses carriage return.

If using a CYBER 18-20 Computer, the operator presses ESCAPE, types J20@, and presses carriage return. This clears program protect, signals a carriage return, and reverts to operator mode.

Ready the card reader if the card reader is the installation device.

The operator then proceeds to initialize the execution described in section 3.8. This is followed by library installation described in section 3.9. If verification of the augmented system is desired, verify the system (in whole or part) using the procedures of section 4.

MSOS now is ready to operate in its augmented form.

A user who does not have RPG II in the MSOS originally ordered from Control Data may add this product to his system. To do this, the user must first order the installation materials (see the MSOS Version 5 Ordering Bulletin). It is assumed that the user has a file manager in the system and that the main memory and mass memory requirements outlined in the MSOS Version 5 Ordering Bulletin are satisfied. Binary copies of each RPG II program are sent to the user.

The user must modify SYSDAT and generate a new system skeleton. A new installation file is then created using LIBILD. This installation file is used to build the augmented system.

8.1 SYSDAT MODIFICATION

Add the code in figure 8-1 at any convenient point in SYSDAT after the job processor file parameters and before the presets.

Add the following presets to the table of presets at the end of SYSDAT:

```
*    SPC   2
          RPG II DIRECTORY FILE NO PRESET
     SPC   2
     ALF   3, RPFDIR
     ADC   RPFDIR
     SPC   2
          RPG II SWITCH STATUS PRESET
     SPC   2
     ALF   3, R9SWCH
     ADC   R9SWCH
```

8.2 SYSTEM SKELETON MODIFICATION

The skeleton records to load RPG in the program library must be added to the skeleton somewhere after the *LIBEDT record and before the two *Z records at the end

```
*          EJT
*               MISCELLANEOUS INFORMATION
*
*               RPGII ENTRIES
*
           SPC   2
*               FILE NUMBER ASSIGNMENT FOR RPGII USAGE
           SPC   2
RPGNBR     EQU   RPGNBR(20)        NO OF FILE NUMBERS RESERVED FOR RPGII
*
RPGTOP     EQU   RPGTOP(FBASV4-1)                  HIGHEST RPGII FILE NO
*
RPGBAS     EQU   RPGBAS(RPGTOP-RPGNBR+1)           LOWEST RPGII FILE NO
*
           ENT   RPGBAS, RPGTOP, RPGNBR
*
           SPC   2
*               FILE NO OF RPGII DIRECTORY STORED BY DFUT - INITIALIZE
           SPC   2
           ENT   RPGDIR
RPGDIR     NUM   0                 FILE NO OF RPGII DIRECTORY
           SPC   2
*               STATUS OF RPGII SWITCHES (EXTERNAL INDICATORS) U1 TO U8
           SPC   2
           ENT   R9SWCH
R9SWCH     NUM   0                 STATUS OF RPGII SWITCHES U1 TO U8
```

Figure 8-1. Partial SYSDAT Modification for Adding RPG II

of the skeleton. Care must be taken to insert these records so that current system modules linked together on mass memory are not interrupted by the insertions. The skeleton records necessary to load RPG in the program library follow. The deck identification field is optional (refer to appendix O). When inserting the skeleton records, the value of n is the logical unit of the installation device. For example, the record *K,In would be inserted as *K,I6 if 6 is the logical unit number of the installation device.

```
*K,In
*LIBEDT
*L,RPGII
*R  'RPGII'    '  DECK-ID R02  RPGII 1.0'
*K,P8
*P,F,,R9BASE
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*T
*K,I8
*N,RPGSM0...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIH'   '  DECK-ID C01  RPGII 1.0'
*T
*K,I8
*N,RPGSM1...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIF'   '  DECK-ID C02  RPGII 1.0'
*T
*K,I8
*N,RPGSM2...8
*K,I6
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIF'   '  DECK-ID C03  RPGII 1.0'
*T
*K,I8
*N,RPGSM3...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIL'   '  DECK-ID C04  RPGII 1.0'
*T
*K,I8
*N,RPGSM4...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIII'   '  DECK-ID C05  RPGII 1.0'
*T
*K,I8
*N,RPGSM5...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIC'   '  DECK-ID C06  RPGII 1.0'
*T
*K,I8
*N,RPGSM6...8
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
```

```
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIO'   '  DECK-ID C07  RPGII 1.0'
*T
*K,I8
*N,RPGSM7,,,B
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIA'   '  DECK-ID C08  RPGII 1.0'
*T
*K,I8
*N,RPGSM8,,,B
*K,In
*P,F,,OVLYPT
*R  'RPG'      '  DECK-ID R01  RPGII 1.0'
*R  'RPGDMY'   '  DECK-ID C11  RPGII 1.0'
*R  'R9LBY'    '  DECK-ID R55  RPGII 1.0'
*R  'R9SBY'    '  DECK-ID R97  RPGII 1.0'
*R  'R9MIW'    '  DECK-ID R63  RPGII 1.0'
*R  'R9MVW'    '  DECK-ID R73  RPGII 1.0'
*R  'R9MIB'    '  DECK-ID R62  RPGII 1.0'
*R  'R9MVB'    '  DECK-ID R71  RPGII 1.0'
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*R  'RPGROT'   '  DECK-ID C10  RPGII 1.0'
*R  'RPGIIM'   '  DECK-ID C09  RPGII 1.0'
*T
*K,I8
*N,RPGSM9,,,B
*K,In
*L,R9ADSB
*R  'R9ADSB'   '  DECK-ID R03  RPGII 1.0'
*L,R9ARG
*R  'R9ARG'    '  DECK-ID R04  RPGII 1.0'
*L,R9ASQR
*R  'R9ASQR'   '  DECK-ID R05  RPGII 1.0'
*L,R9ATON
*R  'R9ATON'   '  DECK-ID R06  RPGII 1.0'
*L,R9BDPR
*R  'R9BDPR'   '  DECK-ID R07  RPGII 1.0'
*L,R9BINP
*R  'R9BINP'   '  DECK-ID R08  RPGII 1.0'
*L,R9BINT
*R  'R9BINT'   '  DECK-ID R09  RPGII 1.0'
*L,R9BITF
*R  'R9BITF'   '  DECK-ID R10  RPGII 1.0'
*L,R9BITN
*R  'R9BITN'   '  DECK-ID R11  RPGII 1.0'
*L,R9CALC
*R  'R9CALC'   '  DECK-ID R12  RPGII 1.0'
*L,R9CHAN
*R  'R9CHAN'   '  DECK-ID R13  RPGII 1.0'
*L,R9CHIN
*R  'R9CHIN'   '  DECK-ID R14  RPGII 1.0'
*L,R9CLOS
*R  'R9CLOS'   '  DECK-ID R15  RPGII 1.0'
*L,R9CLRC
*R  'R9CLRC'   '  DECK-ID R16  RPGII 1.0'
*L,R9CLRE
*R  'R9CLRE'   '  DECK-ID R17  RPGII 1.0'
*L,R9CMBD
*R  'R9CMBD'   '  DECK-ID R18  RPGII 1.0'
*L,R9CMOV
*R  'R9CMOV'   '  DECK-ID R19  RPGII 1.0'
*L,R9CNTR
*R  'R9CNTR'   '  DECK-ID R20  RPGII 1.0'
*L,R9COMP
*R  'R9COMP'   '  DECK-ID R21  RPGII 1.0'
*L,R9CRIN
*R  'R9CRIN'   '  DECK-ID R22  RPGII 1.0'
*L,R9DEBG
*R  'R9DEBG'   '  DECK-ID R23  RPGII 1.0'
*L,R9DETP
*R  'R9DETP'   '  DECK-ID R24  RPGII 1.0'
*L,R9DIVD
*R  'R9DIVD'   '  DECK-ID R25  RPGII 1.0'
*L,R9DMND
*R  'R9DMND'   '  DECK-ID R26  RPGII 1.0'
*L,R9DSPY
*R  'R9DSPY'   '  DECK-ID R27  RPGII 1.0'
*L,R9ECOD
*R  'R9ECOD'   '  DECK-ID R28  RPGII 1.0'
*L,R9EDCN
*R  'R9EDCN'   '  DECK-ID R29  RPGII 1.0'
*L,R9EXCP
*R  'R9EXCP'   '  DECK-ID R30  RPGII 1.0'
*L,R9EXIT
*R  'R9EXIT'   '  DECK-ID R31  RPGII 1.0'
*L,R9FILR
*R  'R9FILR'   '  DECK-ID R32  RPGII 1.0'
*L,R9FINS
*R  'R9FINS'   '  DECK-ID R33  RPGII 1.0'
*L,R9FLDL
*R  'R9FLDL'   '  DECK-ID R34  RPGII 1.0'
*L,R9FLOW
*R  'R9FLOW'   '  DECK-ID R35  RPGII 1.0'
*L,R9FNBG
*R  'R9FNBG'   '  DECK-ID R36  RPGII 1.0'
*L,R9FORC
*R  'R9FORC'   '  DECK-ID R37  RPGII 1.0'
*L,R9FSTL
*R  'R9FSTL'   '  DECK-ID R38  RPGII 1.0'
*L,R9FTOV
*R  'R9FTOV'   '  DECK-ID R39  RPGII 1.0'
*L,R9GETS
*R  'R9GETS'   '  DECK-ID R40  RPGII 1.0'
*L,R9GOTO
*R  'R9GOTO'   '  DECK-ID R41  RPGII 1.0'
*L,R9ICTL
*R  'R9ICTL'   '  DECK-ID R42  RPGII 1.0'
*L,R9IDMX
*R  'R9IDMX'   '  DECK-ID R43  RPGII 1.0'
*L,R9INDM
*R  'R9INDM'   '  DECK-ID R44  RPGII 1.0'
*L,R9INIG
*R  'R9INIG'   '  DECK-ID R45  RPGII 1.0'
*L,R9INIT
*R  'R9INIT'   '  DECK-ID R46  RPGII 1.0'
*L,R9INMV
*R  'R9INMV'   '  DECK-ID R47  RPGII 1.0'
*L,R9INTA
*R  'R9INTA'   '  DECK-ID R48  RPGII 1.0'
```

```
*L,R9INTL
*R  'R9INTL'  '  DECK-ID R49  RPGII 1.0'
*L,R9IOCL
*R  'R9IOCL'  '  DECK-ID R50  RPGII 1.0'
*L,R9IPBG
*R  'R9IPBG'  '  DECK-ID R51  RPGII 1.0'
*L,R9IPUT
*R  'R9IPUT'  '  DECK-ID R52  RPGII 1.0'
*L,R9ITLP
*R  'R9ITLP'  '  DECK-ID R53  RPGII 1.0'
*L,R9LAHD
*R  'R9LAHD'  '  DECK-ID R54  RPGII 1.0'
*L,R9LBY
*R  'R9LBY'   '  DECK-ID R55  RPGII 1.0'
*L,R9LCAE
*R  'R9LCAE'  '  DECK-ID R56  RPGII 1.0'
*L,R9LEL
*R  'R9LEL'   '  DECK-ID R57  RPGII 1.0'
*L,R9LKUP
*R  'R9LKUP'  '  DECK-ID R58  RPGII 1.0'
*L,R9LOAD
*R  'R9LOAD'  '  DECK-ID R59  RPGII 1.0'
*L,R9LOCL
*R  'R9LOCL'  '  DECK-ID R60  RPGII 1.0'
*L,R9LRCK
*R  'R9LRCK'  '  DECK-ID R61  RPGII 1.0'
*L,R9MIB
*R  'R9MIB'   '  DECK-ID R62  RPGII 1.0'
*L,R9MIW
*R  'R9MIW'   '  DECK-ID R63  RPGII 1.0'
*L,R9MMOV
*R  'R9MMOV'  '  DECK-ID R64  RPGII 1.0'
*L,R9MOVA
*R  'R9MOVA'  '  DECK-ID R65  RPGII 1.0'
*L,R9MOVE
*R  'R9MOVE'  '  DECK-ID R66  RPGII 1.0'
*L,R9MOVZ
*R  'R9MOVZ'  '  DECK-ID R67  RPGII 1.0'
*L,R9MTRN
*R  'R9MTRN'  '  DECK-ID R68  RPGII 1.0'
*L,R9MTWK
*R  'R9MTWK'  '  DECK-ID R69  RPGII 1.0'
*L,R9MULT
*R  'R9MULT'  '  DECK-ID R70  RPGII 1.0'
*L,R9MVB
*R  'R9MVB'   '  DECK-ID R71  RPGII 1.0'
*L,R9MVTA
*R  'R9MVTA'  '  DECK-ID R72  RPGII 1.0'
*L,R9MVW
*R  'R9MVW'   '  DECK-ID R73  RPGII 1.0'
*L,R9NRMX
*R  'R9NRMX'  '  DECK-ID R74  RPGII 1.0'
*L,R9NSQR
*R  'R9NSQR'  '  DECK-ID R75  RPGII 1.0'
*L,R9NTOA
*R  'R9NTOA'  '  DECK-ID R76  RPGII 1.0'
*L,R9NXFL
*R  'R9NXFL'  '  DECK-ID R77  RPGII 1.0'
*L,R9NXRC
*R  'R9NXRC'  '  DECK-ID R78  RPGII 1.0'
*L,R9OPNF
*R  'R9OPNF'  '  DECK-ID R79  RPGII 1.0'
*L,R9OTMG
*R  'R9OTMG'  '  DECK-ID S22  RPGII 1.0'
*L,R9OTMV
*R  'R9OTMV'  '  DECK-ID R80  RPGII 1.0'
*L,R9OTPT
*R  'R9OTPT'  '  DECK-ID R81  RPGII 1.0'
*L,R9OV50
*R  'R9OV50'  '  DECK-ID R83  RPGII 1.0'
*L,R9OVOP
*R  'R9OVOP'  '  DECK-ID R82  RPGII 1.0'
*L,R9PACK
*R  'R9PACK'  '  DECK-ID R84  RPGII 1.0'
*L,R9PAGE
*R  'R9PAGE'  '  DECK-ID R85  RPGII 1.0'
*L,R9POSS
*R  'R9POSS'  '  DECK-ID R86  RPGII 1.0'
*L,R9PRCL
*R  'R9PRCL'  '  DECK-ID R87  RPGII 1.0'
*L,R9PRFN
*R  'R9PRFN'  '  DECK-ID R88  RPGII 1.0'
*L,R9PUTS
*R  'R9PUTS'  '  DECK-ID R89  RPGII 1.0'
*L,R9RCAD
*R  'R9RCAD'  '  DECK-ID R90  RPGII 1.0'
*L,R9RDEN
*R  'R9RDEN'  '  DECK-ID R91  RPGII 1.0'
*L,R9READ
*R  'R9READ'  '  DECK-ID R92  RPGII 1.0'
*L,R9RPGO
*R  'R9RPGO'  '  DECK-ID R93  RPGII 1.0'
*L,R9RPRT
*R  'R9RPRT'  '  DECK-ID R94  RPGII 1.0'
*L,R9RSLT
*R  'R9RSLT'  '  DECK-ID R95  RPGII 1.0'
*L,R9RSTS
*R  'R9RSTS'  '  DECK-ID R96  RPGII 1.0'
*L,R9SBY
*R  'R9SBY'   '  DECK-ID R97  RPGII 1.0'
*L,R9SETF
*R  'R9SETF'  '  DECK-ID R98  RPGII 1.0'
*L,R9SETN
*R  'R9SETN'  '  DECK-ID R99  RPGII 1.0'
*L,R9SHFT
*R  'R9SHFT'  '  DECK-ID S01  RPGII 1.0'
*L,R9SKIP
*R  'R9SKIP'  '  DECK-ID S02  RPGII 1.0'
*L,R9SPAC
*R  'R9SPAC'  '  DECK-ID S03  RPGII 1.0'
*L,R9SPTP
*R  'R9SPTP'  '  DECK-ID S04  RPGII 1.0'
*L,R9SQRT
*R  'R9SQRT'  '  DECK-ID S05  RPGII 1.0'
*L,R9STH0
*R  'R9STH0'  '  DECK-ID S06  RPGII 1.0'
*L,R9STLL
*R  'R9STLL'  '  DECK-ID S07  RPGII 1.0'
*L,R9STON
*R  'R9STON'  '  DECK-ID S08  RPGII 1.0'
*L,R9TBOT
*R  'R9TBOT'  '  DECK-ID S09  RPGII 1.0'
*L,R9TIME
*R  'R9TIME'  '  DECK-ID S10  RPGII 1.0'
*L,R9TP40
*R  'R9TP40'  '  DECK-ID S11  RPGII 1.0'
*L,R9TRAL
*R  'R9TRAL'  '  DECK-ID S12  RPGII 1.0'
*L,R9TRCE
*R  'R9TRCE'  '  DECK-ID S13  RPGII 1.0'
*L,R9TROT
*R  'R9TROT'  '  DECK-ID S14  RPGII 1.0'
*L,R9TSTB
*R  'R9TSTB'  '  DECK-ID S15  RPGII 1.0'
*L,R9TSTN
*R  'R9TSTN'  '  DECK-ID S16  RPGII 1.0'
*L,R9TSTZ
*R  'R9TSTZ'  '  DECK-ID S17  RPGII 1.0'
```

```
*L,R9TTOP
*A  'R9TTOP'  '  DECK-ID S18  RPGII 1.0'
*L,R9UNPK
*A  'R9UNPK'  '  DECK-ID S19  RPGII 1.0'
*L,R9XCPT
*A  'R9XCPT'  '  DECK-ID S20  RPGII 1.0'
*L,R9XFOT
*A  'R9XFOT'  '  DECK-ID S21  RPGII 1.0'
*L,R9XRSD
*A  'R9XRSD'  '  DECK-ID S23  RPGII 1.0'
*L,R9YCOD
*A  'R9YCOD'  '  DECK-ID S24  RPGII 1.0'
*L,R9ZADS
*A  'R9ZADS'  '  DECK-ID S25  RPGII 1.0'
*L,R9ZCOD
*A  'R9ZCOD'  '  DECK-ID S26  RPGII 1.0'
*L,DFUT
*A  'DFUT'    '  DECK-ID U12  RPGII 1.0'
*P,F
*A  'DFUT'    '  DECK-ID U12  RPGII 1.0'
*A  'DUE'     '  DECK-ID U13  RPGII 1.0'
*A  'DECHEX'  '  DECK-ID U11  RPGII 1.0'
*A  'BINASC'  '  DECK-ID U03  RPGII 1.0'
*A  'ATOM'    '  DECK-ID U02  RPGII 1.0'
*A  'FATMGR'  '  DECK-ID U14  RPGII 1.0'
*A  'MOVBYT'  '  DECK-ID U46  RPGII 1.0'
*A  'CMSG'    '  DECK-ID U37  RPGII 1.0'
*A  'HELP'    '  DECK-ID U15  RPGII 1.0'
*A  'MM'      '  DECK-ID U17  RPGII 1.0'
*A  'MOTION'  '  DECK-ID U18  RPGII 1.0'
*A  'BINHEX'  '  DECK-ID U04  RPGII 1.0'
*A  'ERRMSG'  '  DECK-ID U42  RPGII 1.0'
*A  'DEFINE'  '  DECK-ID U39  RPGII 1.0'
*A  'AUDIT'   '  DECK-ID U32  RPGII 1.0'
*A  'DUMP'    '  DECK-ID U41  RPGII 1.0'
*A  'INIT'    '  DECK-ID U44  RPGII 1.0'
*A  'LOAD'    '  DECK-ID U45  RPGII 1.0'
*A  'COPY'    '  DECK-ID U38  RPGII 1.0'
*A  'DISCRD'  '  DECK-ID U40  RPGII 1.0'
*A  'ADROUT'  '  DECK-ID U30  RPGII 1.0'
*A  'PURGE'   '  DECK-ID U47  RPGII 1.0'
*A  'SAVE'    '  DECK-ID U51  RPGII 1.0'
*A  'RELOAD'  '  DECK-ID U50  RPGII 1.0'
*A  'TAPMGR'  '  DECK-ID U22  RPGII 1.0'
*A  'ADRPRG'  '  DECK-ID U31  RPGII 1.0'
*A  'ADRSKL'  '  DECK-ID U01  RPGII 1.0'
*A  'FORTN'   '  DECK-ID U54  RPGII 1.0'
*A  'Q8PRMS'  '  DECK-ID U55  RPGII 1.0'
*T
*K,I8
*N,DISKUT,,,B
*K,In
*L,RPGFIL
*A  'RPGFIL'  '  DECK-ID U20  RPGII 1.0'
*L,CATLOG
*A  'CATLOG'  '  DECK-ID U06  RPGII 1.0'
*K,P8
*P,F,,CATSEG
*A  'CATLOG'  '  DECK-ID U06  RPGII 1.0'
*A  'CATFIL'  '  DECK-ID U34  RPGII 1.0'
*A  'CATGET'  '  DECK-ID U05  RPGII 1.0'
*A  'CATSKL'  '  DECK-ID U07  RPGII 1.0'
*A  'CATOBJ'  '  DECK-ID U36  RPGII 1.0'
*A  'CATERR'  '  DECK-ID U33  RPGII 1.0'
*A  'CATMSG'  '  DECK-ID U35  RPGII 1.0'
*A  'FORTN'   '  DECK-ID U54  RPGII 1.0'
*A  'Q8PRMS'  '  DECK-ID U55  RPGII 1.0'
*T
*K,I8
*N,CATFIL,,,B
*K,In
*L,SWITCH
*A  'SWITCH'  '  DECK-ID U21  RPGII 1.0'
*K,P8
*P,F,,SWISEG
*A  'SWITCH'  '  DECK-ID U21  RPGII 1.0'
*A  'SWIFIL'  '  DECK-ID U53  RPGII 1.0'
*A  'FORTN'   '  DECK-ID U54  RPGII 1.0'
*A  'Q8PRMS'  '  DECK-ID U55  RPGII 1.0'
*T
*K,I8
*N,SWIFIL,,,B
*K,In
*L,RBDPCH
*A  'RBDPCH'  '  DECK-ID U24  RPGII 1.0'
*K,P8
*P,F,,RBDSEG
*A  'RBDPCH'  '  DECK-ID U24  RPGII 1.0'
*A  'RBDFIL'  '  DECK-ID U48  RPGII 1.0'
*A  'RBPERR'  '  DECK-ID U49  RPGII 1.0'
*A  'GETLPG'  '  DECK-ID U43  RPGII 1.0'
*A  'SECGET'  '  DECK-ID U52  RPGII 1.0'
*A  'FORTN'   '  DECK-ID U54  RPGII 1.0'
*A  'Q8PRMS'  '  DECK-ID U55  RPGII 1.0'
*T
*K,I8
*N,RBDFIL,,,B
*K,In
*L,OPEN01
*A  'OPEN01'  '  DECK-ID M01  RPGII 1.0'
*L,OPEN02
*A  'OPEN02'  '  DECK-ID M02  RPGII 1.0'
*L,OPEN03
*A  'OPEN03'  '  DECK-ID M03  RPGII 1.0'
*L,OPEN04
*A  'OPEN04'  '  DECK-ID M04  RPGII 1.0'
*L,OPEN05
*A  'OPEN05'  '  DECK-ID M05  RPGII 1.0'
*L,OPEN06
*A  'OPEN06'  '  DECK-ID M06  RPGII 1.0'
*L,OPEN07
*A  'OPEN07'  '  DECK-ID M07  RPGII 1.0'
*L,READ09
*A  'READ09'  '  DECK-ID M09  RPGII 1.0'
*L,READ10
*A  'READ10'  '  DECK-ID M10  RPGII 1.0'
*L,READ11
*A  'READ11'  '  DECK-ID M11  RPGII 1.0'
*L,READ12
*A  'READ12'  '  DECK-ID M12  RPGII 1.0'
*L,READ13
*A  'READ13'  '  DECK-ID M13  RPGII 1.0'
*L,READ14
*A  'READ14'  '  DECK-ID M14  RPGII 1.0'
*L,READ15
*A  'READ15'  '  DECK-ID M15  RPGII 1.0'
*L,WRIT17
*A  'WRIT17'  '  DECK-ID M17  RPGII 1.0'
*L,WRIT18
*A  'WRIT18'  '  DECK-ID M18  RPGII 1.0'
*L,WRIT19
*A  'WRIT19'  '  DECK-ID M19  RPGII 1.0'
*L,WRIT20
*A  'WRIT20'  '  DECK-ID M20  RPGII 1.0'
*L,WRIT21
*A  'WRIT21'  '  DECK-ID M21  RPGII 1.0'
*L,WRIT22
*A  'WRIT22'  '  DECK-ID M22  RPGII 1.0'
```

```
*L,WRIT23
*R  'WRIT23'  '  DECK-ID M23  RPGII 1.0'
*L,UPDT25
*R  'UPDT25'  '  DECK-ID M25  RPGII 1.0'
*L,UPDT26
*R  'UPDT26'  '  DECK-ID M26  RPGII 1.0'
*L,UPDT27
*R  'UPDT27'  '  DECK-ID M27  RPGII 1.0'
*L,UPDT28
*R  'UPDT28'  '  DECK-ID M28  RPGII 1.0'
*L,UPDT29
*R  'UPDT29'  '  DECK-ID M29  RPGII 1.0'
*L,ADDT30
*R  'ADDT30'  '  DECK-ID M30  RPGII 1.0'
*L,ADDT31
*R  'ADDT31'  '  DECK-ID M31  RPGII 1.0'
*L,ADDT32
*R  'ADDT32'  '  DECK-ID M32  RPGII 1.0'
*L,SETL33
*R  'SETL33'  '  DECK-ID M33  RPGII 1.0'
*L,SLCT34
*R  'SLCT34'  '  DECK-ID M34  RPGII 1.0'
*L,CLOS35
*R  'CLOS35'  '  DECK-ID M35  RPGII 1.0'
*L,CLOS36
*R  'CLOS36'  '  DECK-ID M36  RPGII 1.0'
*L,CLOS37
*R  'CLOS37'  '  DECK-ID M37  RPGII 1.0'
*L,CLOS38
*R  'CLOS38'  '  DECK-ID M38  RPGII 1.0'
*L,CLOS39
*R  'CLOS39'  '  DECK-ID M39  RPGII 1.0'
*L,CLOS40
*R  'CLOS40'  '  DECK-ID M40  RPGII 1.0'
*L,CLOS41
*R  'CLOS41'  '  DECK-ID M41  RPGII 1.0'
*L,ROOT43
*R  'ROOT43'  '  DECK-ID M43  RPGII 1.0'
*L,ROOT44
*R  'ROOT44'  '  DECK-ID M44  RPGII 1.0'
*L,ROOT45
*R  'ROOT45'  '  DECK-ID M45  RPGII 1.0'
*L,ERR046
*R  'ERR046'  '  DECK-ID M46  RPGII 1.0'
*L,NTAP48
*R  'NTAP48'  '  DECK-ID M48  RPGII 1.0'
*L,NTAP49
*R  'NTAP49'  '  DECK-ID M49  RPGII 1.0'
*L,SORT50
*R  'SORT50'  '  DECK-ID M50  RPGII 1.0'
*L,CKEY51
*R  'CKEY51'  '  DECK-ID M51  RPGII 1.0'
*L,NTAP52
*R  'NTAP52'  '  DECK-ID M52  RPGII 1.0'
*L,NTAP53
*R  'NTAP53'  '  DECK-ID M53  RPGII 1.0'
*L,NTAP54
*R  'NTAP54'  '  DECK-ID M54  RPGII 1.0'
*L,NTAP55
*R  'NTAP55'  '  DECK-ID M55  RPGII 1.0'
*L,NTAP57
*R  'NTAP57'  '  DECK-ID M57  RPGII 1.0'
*L,NTAP58
*R  'NTAP58'  '  DECK-ID M58  RPGII 1.0'
*L,MOUNT
*R  'MOUNT'   '  DECK-ID U19  RPGII 1.0'
*L,IOOOWR
*R  'IOOOWR'  '  DECK-ID U16  RPGII 1.0'
*L,CM00PT
*R  'CM00PT'  '  DECK-ID U08  RPGII 1.0'
*L,CM02IN
*R  'CM02IN'  '  DECK-ID U09  RPGII 1.0'
*L,CM03GO
*R  'CM03GO'  '  DECK-ID U10  RPGII 1.0'
*L,STRACE
*R  'STRACE'  '  DECK-ID M59  RPGII 1.0'
```

It may be necessary to decrease the value of N4, the size of allocatable area 4, in order to increase the size of unprotected main memory to satisfy the RPG requirements for unprotected memory. RPG requires an unprotected area of at least 18,270 bytes for compilation. RPG execution requirements for unprotected are as follows:

Size	Under MSOS Load and Go	Under MSOS Catalog Mode
Minimum:	14,000 bytes	12,000 bytes
Typical:	36,000 bytes	30,000 bytes
Large programs:	56,000 bytes	40,000 bytes

The current size of unprotected may be ascertained by dumping the contents of the locations $F6_{16}$ and $F7_{16}$ and computing the difference.

$V = F6_{16} - F7_{16}$ = number of words of unprotected main memory

Let V' equal the new size of unprotected main memory. Then:

V' equals V - 13

This is because 13 words were added to SYSDAT.

If the value of V' is less than the required size of unprotected, it may be possible to allow more space in unprotected by decreasing the value of N4. The requirements for N4 are discussed in appendix M. To change N4, modify the skeleton record *S, N4, n so that n is the new value of N4. This record is near the beginning of the skeleton.

8.3 INCORPORATING CHANGES INTO SYSTEM

A new installation file must now be created using the new binary version of SYSDAT, the new skeleton, the old installation file, and the RPG binaries. This is accomplished by using LIBILD. Care must be taken that the new version of SYSDAT is read by LIBILD before reading the old installation file, so that the proper version of SYSDAT will be incorporated into the new installation file.

Using the new installation file and the system initializer program, the new system may be loaded. Entries on the comment device are as follows:

Entry	Remarks
MI	Operator has manually interrupted the system
*BATCH	Operator requests batch processing
J	Batch processing is in control
*JOB	Operator requests the job processor
J	Job processor is in control
*SILP	Operator requests the system initializer loading program

The system replies:

THE INITIALIZER WILL BE MOVED TO LOCATION xxxx AND EXECUTED. TURN OFF PROTEC SWITCH AND TYPE CARRIAGE RETURN

If using a 1700 Series computer system, the operator sets the protect switch to the neutral position and presses carriage return.

If using a CYBER 18-20 computer, the operator presses ESCAPE, types J20@, and presses carriage return. This clears program protect, signals a carriage return, and reverts to operator mode.

Ready the card reader if the card reader is the installation device.

The operator then proceeds to the initializer execution described in section 3.8. This is followed by the library installation described in section 3.9. If verification of the augmented system is desired, verify the system (in whole or part) using the procedures of section 4.

MSOS is now ready to operate in its augmented form.

A user who does not have the macro assembler in the version of MSOS originally ordered from Control Data may add this product to his system. To do this, he must order the installation materials (see the MSOS Version 5 Ordering Bulletin). The macro assembler installation materials, on punched cards or magnetic tape, consist of a binary copy of each macro assembler program.

Since SYSDAT is not modified by this addition, the user need not rebuild the entire system using the techniques of section 5, method 1. Instead, by using the techniques of section 5, method 2, the user may generate a separate installation file for this product alone. This file is used to update the system library, thereby adding the new product to the system.

Using method 2, the installation steps are:

1. Use SKED utility to produce the macro assembler skeleton file. Records that compose the file are specified in this section.
2. Use the LIBILD utility to produce the macro assembler installation file, a complete and independent installation file containing the skeleton and binary programs for the macro assembler alone.
3. Use the LIBEDT utility to enter the macro assembler programs from the new installation file into the program library.
4. (Optional) The system may be verified in whole or in part using the verification procedures described in section 4.

The detailed procedures for steps 1, 2, and 3 are described below.

9.1 BUILDING MACRO ASSEMBLER INSTALLATION FILE SKELETON

If the system has a card reader, the skeleton records defined in figure 9-1 may be punched and used as the LIBILD skeleton input (described in section 9.2).

If no card reader is available, the skeleton records can be put onto magnetic tape using the following MSOS functions. The operator enters the job processor with:

```
*JOB
```

After the system replies with:

```
J
```

the operator calls the library editor with:

```
*LIBEDT
```

to transfer the first two records of the skeleton to an output tape mounted on logical unit 6. After the library editor is loaded, transfer of records is accomplished by:

*T, 4, A, 6, A, 2	Transfers two records from the comment device to logical unit 6 (ASCII mode)
*K, I6, P8 *L, LIBMAC	First two records from figure 9-1 (macro-assembler skeleton records)
*Z *Z	Exit LIBEDT Exit job processor Control statements; not transferred to tape.

Next, the operator calls debug to close the file and to rewind the new tape:

```
MI
DB
```

When the system replies that debug is loaded (DEBUG IN), the operator writes an end-of-file mark and rewind the tape:

WEF, 6, 1	Write end-of-file mark.
NEXT	
REW, 6	Rewind tape.
NEXT	
OFF	Exit from debug.

The skeleton editor can now be used to build the remainder of the installation file. After entering the job processor, SKED is called with:

```
*SKED
```

The computer replies with:

```
SKED IN
NEXT
```

OPERATOR ENTERED CODE (SKELETON)	SKELETON DIRECTS PROCESSING AS SHOWN:
*K, I6, P8	ALL INPUT IS FOR LOGICAL UNIT 6. IF THE INSTALLATION IS TO BE MADE FROM ANOTHER LOGICAL UNIT, CHANGE I VALUE AS APPROPRIATE.
*L, LIBMAC	
*B 'LIBMAC'	
*L.ASSEM	
*B 'ASSEM'	
*K. P8	
*P. F	SET OF BINARY PROGRAMS IS ENTERED WITH MSOS PROGRAM LIBRARY AS AN ABSOLUTE FILE.
*B 'PASS1'	
*B 'PA1PR2'	
*T	
*K. I8	
*N. PASS1...8	
*K. I6	
*K. P8	
*P. F	ABSOLUTE FILE
*B 'PASS2'	
*B 'PA2PR2'	
*T	
*K. I8	
*N, PASS2... B	
*K, I6	
*K, P8	
*P, F	ABSOLUTE FILE
*B 'PASS3'	
*B 'PA3PR2'	
*B 'PA3PR3'	
*T	
*K. I8	
*N. PASS3... B	
*K. I6	
*K. P8	
*P. F	ABSOLUTE FILE
*B 'TABLST'	
*T	
*K. I8	
*N. TABLST... B	
*K. I6	
*K. P8	
*P. F	ABSOLUTE FILE
*B 'XREF'	
*K. I8	
*N. XREF... B	
*K. I6	THESE TWO PROGRAMS REMAIN IN BINARY FORMAT.
*B 'MACSKL'	
*B 'MACROS'	
*Z	EXIT LIBEDT / EXIT JOB — CONTROL STATEMENTS USED DURING LIBEDT PROCESSING
*CT0, MACRO ASSEMBLER INSTALLED	
*Z	

NOTE: EACH LINE ENDS WITH A CARRIAGE RETURN AND A LINE FEED.

Figure 9-1. Skeleton for Adding Macro Assembler to System

The operator loads the first two records just written on the installation tape:

```
LOAD, 6
```

After loading the records, the computer replies:

```
ANY MORE INPUT, ENTER LU
```

This SKED internal condition is cleared by pressing carriage return. The computer replies:

```
NEXT
```

The operator now inserts all the other records in the skeleton of figure 9-1. Each input record is followed by a carriage return and a line feed.

```
INSERT, 2, 4
*B   'LIBMAC'
*L, ASSEM
.
.
.
*Z
carriage return
```

Remaining records from skeleton in figure 9-1

The final carriage return terminates the loading command sequence. The comment device displays:

```
NEXT
```

The operator may now list the full skeleton by:

```
CATLOG
```

When the computer replies:

```
NEXT
```

the operator is ready to dump the skeleton onto the installation tape, using the same magnetic tape as before:

```
REW, 6
```

When the tape is rewound and the computer replies (NEXT), the operator writes the skeleton with:

```
DUMP, 6
```

The computer replies with:

```
NEXT
```

and the skeleton is now saved on the output device. The operator exits from the skeleton editor with:

```
EXIT
```

9.2 BUILDING MACRO ASSEMBLER INSTALLATION FILE

The skeleton is now used in conjunction with the macro assembler binaries to create the macro assembler installation file. The utility program LIBILD generates the file. In the following example, both the new skeleton and the macro assembler binaries are on magnetic tape and are input from logical unit 16.

Assuming that the job processor is still in control of the computer, LIBILD is put in control by the operator entering:

```
*LIBILD
```

The Comment Device Displays:	The Operator Replies:
CONTROL LU =	Carriage return
DEFS LU =	Carriage return
INSTALL LU =	Six carriage returns
NEWLIB LU =	Carriage return
LIB 01 LU =	16 carriage returns
LIB 02 LU =	Carriage return
SKELETON LU =	16 carriage returns

Note that only a carriage return is entered as the response to the query CONTROL LU =. This is because the sequence control statements are read from the comment device. A carriage return is also the response to the query DEFS LU =, since the installation file is to be created according to the skeleton and not according to a definitions deck. The response to INSTALL LU = indicates that the installation file is to be written on logical unit 6. No new output library is to be created. Therefore, a carriage return is entered following the query NEWLIB LU =.

Following the 16-carriage-return reply to the query SKELETON LU =, the binary programs are read from logical unit 16 and saved on mass storage. When all the binary programs have been read, the comment device displays:

```
LOAD SKEL/INSTAL, CR WHEN READY
```

After mounting the skeleton tape prepared by SKED (section 9.1) on logical unit 16, mounting the installation tape on logical unit 6, and readying both of these units, the operator replies with a carriage return. LIBILD reads the skeleton and prepares a macro assembly installation file by reading the skeleton records and binaries and processing the binaries according to the instructions of the skeleton records.

When the installation file is complete, the comment device displays:

```
LIBRARY BUILD COMPLETE
TYPE *Z TO TERMINATE OR
TYPE *C TO CONTINUE WITH CURRENT
  SKELETON AND/OR
OUTPUT LIBRARY LU'S
```

Since the installation file is now complete and residing on logical unit 6, the operator exits from LIBILD by replying:

```
Z
```

The system returns to job processor control.

9.3 ENTERING MACRO ASSEMBLER INTO MSOS

The newly prepared macro assembler installation file may now be used by LIBEDT utility to enter the macro assembler into MSOS.

Assuming that the job processor is still controlling the computer, the library editor is placed in control by the operator entering:

```
*LIBEDT
```

When the library editor is controlling the computer, the comment device displays:

```
LIB IN
```

The operator loads the installation file on logical unit 6 and readies that tape unit. He then causes the file to be read by entering:

```
*V,6
```

After LIBEDT has processed the file, the comment device displays the CTO statement entered at the end of the skeleton:

```
MACRO ASSEMBLER INSTALLED
```

The first *Z statement from the skeleton causes LIBEDT termination; the second *Z statement causes job processor termination. MSOS is now augmented by addition of the macro assembler.

A user who does not have Sort/Merge in the version of MSOS originally ordered from Control Data may add this product to his system. To do this, he must order the installation materials (see the MSOS Version 5 Ordering Bulletin).

It is assumed the user's system contains a file manager. It is further assumed that the user has in his system the main memory and mass memory requirements for Sort/Merge Version 1.0 as outlined in the MSOS Version 5 Ordering Bulletin. The sort/merge installation materials, on punched cards or magnetic tape, consist of a binary copy of each sort/merge program.

Since SYSDAT is not modified by this addition, the user need not rebuild the entire system using the techniques of section 5, method 1. Instead, using the techniques of section 5, method 2, the user may generate a separate installation file for this product alone. That file is used to update the system library, thereby adding the new product to the system.

Using method 2, the installation steps are:

1. Use SKED utility to produce the Sort/Merge skeleton file. Records that compose the file are specified in this section.

2. Use the LIBILD utility to produce the Sort/Merge installation file, a complete and independent installation file containing the skeleton and binary programs for Sort/Merge alone.

3. Use the LIBEDT utility to enter the Sort/Merge programs from the new installation file into the program library.

4. (Optional) The system may be verified in whole or in part using the verification procedures described in section 4.

The detailed procedures for steps 1, 2, and 3 are described below.

10.1 BUILDING SORT/MERGE INSTALLATION FILE SKELETON

If the system has a card reader, the skeleton records defined in figure 10-1 may be punched and used as the LIBILD skeleton input (described in section 10.2).

If no card reader is available, the skeleton records can be put onto magnetic tape using the following MSOS functions. The operator enters the job processor with:

```
*JOB
```

After the system replies with:

```
J
```

the operator calls the library editor with:

```
*LIBEDT
```

to transfer the first two records of the skeleton to an output tape mounted on logical unit 6. After the library editor is loaded, this is accomplished by:

*T, 4, A, 6, A, 2	Transfers two records from the comment device to logical unit 6 (ASCII mode)
*K, I6 *L, SMC	First two records from figure 10-1 (macro-assembler skeleton records)
*Z	Exit LIBEDT
*Z	Exit job processor

Control statements; not transferred as records.

Next, the operator calls debug to close the file and to rewind the new tape:

```
MI
DB
```

When the system replies that debug is loaded (DEBUG IN), the operator writes an end-of-file mark and rewinds the tape:

```
WEF, 6, 1     Write end-of-file mark.
NEXT
REW, 6        Rewind tape.
NEXT
OFF           Exit from debug.
```

The skeleton editor can now be used to build the remainder of the installation file. After entering the job processor, SKED is called with:

```
*SKED
```

OPERATOR ENTERED CODE (SKELETON)	SKELETON DIRECTS PROCESSING AS SHOWN:
*V *V SORT/MERGE 1.0 *V *K, I6 *L, SMC *B 'SMC' ' DECK-ID S01 SMC 1.0' *K, P8 *P *B 'SMCMON' ' DECK-ID S02 SMC 1.0' *B 'FLOTN' ' DECK-ID S08 SMC 1.0' *B 'PARASN' ' DECK-ID S07 SMC 1.0' *B 'COMNER' ' DECK-ID S09 SMC 1.0' *B 'NXTLOC' ' NEXT AVAILABLE LOCATION' *T *K, I8 *N, SMCMON... B	ALL INPUT IS FOR LOGICAL UNIT 6. IF THE INSTALLATION IS TO BE MADE FROM ANOTHER LOGICAL UNIT, CHANGE I VALUE AS APPROPRIATE. SET OF BINARY PROGRAMS IS ENTERED INTO MSOS PROGRAM LIBRARY AS AN ABSOLUTE FILE.
*K, In *P *B 'SMCEDT' ' DECK-ID S03 SMC 1.0' *B 'NXTLOC' ' NEXT AVAILABLE LOCATION' *T *K, I8 *N, SMCEDT... B	ABSOLUTE FILE
*K, In *P *B 'SMCSRT' ' DECK-ID S04 SMC 1.0' *B 'NXTLOC' ' NEXT AVAILABLE LOCATION' *T *K, I8 *N, SMCSRT, ... B	ABSOLUTE FILE
*K, In *P *B 'SMCIMG' ' DECK-ID S05 SMC 1.0' *B 'NXTLOC' ' NEXT AVAILABLE LOCATION' *T *K, I8 *N, SMCIMG... B	ABSOLUTE FILE
*K, In *P *B 'SMCFMG' ' DECK-ID S06 SMC 1.0' *B 'NXTLOC' ' NEXT AVAILABLE LOCATION' *T *K, I8 *N, SMCFMG... B	ABSOLUTE FILE
*Z *CTO, SORT/MERGE INSTALLED *Z	EXIT LIBEDT EXIT JOB CONTROL STATEMENTS USED DURING LIBEDT PROCESSING

NOTE: EACH LINE ENDS WITH A CARRIAGE RETURN AND A LINE FEED. DECK IDENTIFICATIONS ARE OPTIONAL.

Figure 10-1. Skeleton for Adding Sort/Merge to System

The computer replies with:

SKED IN

NEXT

The operator loads the first two records just written on the installation tape:

LOAD, 6

After loading the records, the computer replies:

ANY MORE INPUT, ENTER LU

This SKED internal condition is cleared by pressing carriage return. The computer replies:

NEXT

The operator now inserts all the other records in the skeleton of figure 10-1. Each input record is followed by a carriage return and a line feed.

INSERT, 2, 4

*B 'SMC'
*K, P8
.
.
.
*Z

Remaining records from skeleton in figure 10-1

carriage return

The final carriage return terminates the loading command sequence. The comment device displays:

NEXT

The operator may now list the full skeleton by:

CATLOG

When the computer replies:

NEXT

the operator is ready to dump the skeleton onto the installation tape, using the same magnetic tape as before:

REW, 6

When the tape is rewound and the computer replies (NEXT), the operator writes the skeleton with:

DUMP, 6

The computer replies with:

NEXT

and the skeleton is now saved on the output device. The operator exits from the skeleton editor with:

EXIT

It may be necessary to decrease the value of N4. the size of allocatable area 4, in order to increase the size of unprotected to satisfy the Sort/Merge requirements for unprotected memory. Sort/Merge requires an unprotected area of 12,000 bytes. Speed of execution is improved if additional unprotected memory is available.

The current size of unprotected may be ascertained by dumping the contents of the locations $F6_{16}$ and $F7_{16}$ and computing the difference.

$$V = F6_{16} - F7_{16} = \text{number of words of unprotected.}$$

If the value of V is less than the required 6000 words, it may be possible to allow more space in unprotected by decreasing the value of N4. The requirements for N4 are discussed in appendix M. To change N4, modify the skeleton record *S, N4. n so that n is the new value of N4. This record is near the beginning of the skeleton.

10.2 BUILDING SORT/MERGE INSTALLATION FILE

The skeleton is now used in conjunction with the Sort/Merge binaries to create the sort/merge installation file. The utility program LIBILD generates the file. In the following example. both the new skeleton and the Sort/Merge binaries are on magnetic tape, both mounted on logical unit 16.

Assuming that the job processor is still in control of the computer, LIBILD is put in control by the operator entering:

*LIBILD

The Comment Device Displays:	The Operator Replies:
CONTROL LU =	Carriage return
DEFS LU =	Carriage return
INSTALL LU =	Six carriage returns
NEWLIB LU =	Carriage return
LIB 01 LU =	16 carriage returns
LIB 02 LU =	Carriage return
SKELETON LU =	16 carriage returns

Note that only a carriage return is entered as the response to the query CONTROL LU = . This is because the sequence control statements are read from the comment device. A carriage return is also the response to the query DEFS LU = , since the installation file is to be created according to the skeleton and not according to a definitions deck. The response to INSTALL LU = indicates that the installation file is to be written on logical unit 6. No new output library is to be created. Therefore, a carriage return is entered following the query NEWLIB LU = .

Following the 16-carriage-return reply to the query SKELETON LU = , the binary programs are read from logical unit 16 and saved on mass storage. When all the binary programs have been read, the comment device displays:

```
LOAD SKEL/INSTAL, CR WHEN READY
```

After mounting the skeleton tape prepared by SKED (section 10.1) on logical unit 16, mounting the installation tape on logical unit 6, and readying both of these units, the operator replies with a carriage return. LIBILD reads the skeleton and prepares a Sort/Merge installation file by reading the skeleton records and binaries and processing the binaries according to the instructions of the skeleton records.

When the installation file is complete, the comment device displays:

```
LIBRARY BUILD COMPLETE
TYPE *Z TO TERMINATE OR
TYPE *C TO CONTINUE WITH CURRENT
SKELETON AND/OR
OUTPUT LIBRARY LU'S
```

Since the installation file is now complete and residing on logical unit 6, the operator exits from LIBILD by replying:

```
*Z
```

The system returns to job processor control.

10.3 ENTERING SORT/MERGE INTO MSOS

The newly prepared Sort/Merge installation file may now be used by LIBEDT utility to enter Sort/Merge into MSOS.

Assuming that the job processor is still controlling the computer, the library editor is placed in control by the operator entering:

```
*LIBEDT
```

When the library editor is controlling the computer, the comment device displays:

```
LIB IN
```

The operator loads the installation file on logical unit 6 and readies that tape unit. He then causes the file to be read by entering:

```
*V, 6
```

After LIBEDT has entered the file, the comment device displays the CTO statement entered at the end of the skeleton:

```
SORT/MERGE INSTALLED
```

The first *Z statement from the skeleton causes LIBEDT termination; the second *Z statement causes job processor termination. MSOS is now augmented by addition of Sort/Merge.

A user who does not have the magnetic tape utility processor (MTUP) in the version of MSOS originally ordered from Control Data may add this product to his system. To do this, he must order the installation materials (see the MSOS Version 5 Ordering Bulletin). The magnetic tape utility processor installation material consists of a binary copy of each magnetic tape utility processor program. Since SYSDAT is not modified by this addition, the user need not rebuild the entire system using the techniques of section 5, method 1. Instead, employing the techniques of section 5, method 2, the user may generate a separate installation file for this product alone. That file is used to update the system library, thereby adding the new product to the system.

Using method 2, the installation steps are:

1. Use SKED utility to produce the magnetic tape utility processor skeleton file. Records that compose the file are specified in this section.
2. Use the LIBILD utility to produce the magnetic tape utility processor installation file, a complete and independent installation file containing the skeleton and binary programs for the magnetic tape utilities alone.
3. Use the LIBEDT utility to enter the magnetic tape utility processor programs from the new installation file into the program library.
4. (Optional) The system may be verified in whole or in part using the verification procedures described in section 4.

The detailed procedures for steps 1, 2, and 3 are described below.

11.1 BUILDING MTUP INSTALLATION FILE SKELETON

If the system has a card reader, the skeleton records defined in figure 11-1 may be punched and used as the LIBILD skeleton input (described in section 11.2).

If no card reader is available, the skeleton records can be put onto magnetic tape using the following MSOS functions. The operator enters the job processor with:

```
*JOB
```

After the system replies with:

```
J
```

the operator calls the library editor with:

```
*LIBEDT
```

to transfer the first two records of the skeleton to an output tape mounted on logical unit 6. After the library editor is loaded, this is accomplished by:

*T, 4, A, 6, A, 2	Transfers two records from the comment device to logical unit 6 (ASCII mode)
*K, I6, P8 *L, MTUP	First two records from figure 11-1 (macro-assembler skeleton records)
*Z	Exit LIBEDT — Control statements; not transferred as records
*Z	Exit job processor — Control statements; not transferred as records

Next, the operator calls debug to close the file and to rewind the new tape:

```
MI
DB
```

When the system replies that debug is loaded (DEBUG IN), the operator writes an end-of-file mark and rewinds the tape:

```
WEF, 6, 1      Write end-of-file mark.
NEXT
REW, 6         Rewind tape.
NEXT
OFF            Exit from debug.
```

The skeleton editor can now be used to build the remainder of the installation file. After entering the job processor, SKED is called with:

```
*SKED
```

The computer replies with:

```
SKED IN
NEXT
```

OPERATOR ENTERED CODE (SKELETON)

```
*K, I6, P8
*L, MTUP
*B   'MTUP'      '  DECK-ID  U01   MAG TAPE UTILITY'
*P, F, , TAPUTL
*B   'MTUP'      '  DECK-ID  U01   MAG TAPE UTILITY'
*B   'TAPUTL     '  DECK-ID  U02   MAG TAPE UTILITY'
*B   'FNN'       '  DECK-ID  U03   MAG TAPE UTILITY'
*B   'SCAN'      '  DECK-ID  U04   MAG TAPE UTILITY'
*B   'OPFNIO'    '  DECK-ID  U05   MAG TAPE UTILITY'
*B   'RDWTP'     '  DECK-ID  U06   MAG TAPE UTILITY'
*B   'LIOG'      '  DECK-ID  U07   MAG TAPE UTILITY'
*B   'COPY'      '  DECK-ID  U08   MAG TAPE UTILITY'
*B   'EXIT'      '  DECK-ID  U09   MAG TAPE UTILITY'
*B   'PRINT'     '  DECK-ID  U10   MAG TAPE UTILITY'
*B   'VERIFY'    '  DECK-ID  U11   MAG TAPE UTILITY'
*B   'SELECT'    '  DECK-ID  U12   MAG TAPE UTILITY'
*B   'PRINTT'    '  DECK-ID  U13   MAG TAPE UTILITY'
*B   'DUMP'      '  DECK-ID  U14   MAG TAPE UTILITY'

*B   'INIT'      '  DECK-ID  U15   MAG TAPE UTILITY'
*B   'STNLAB'    '  DECK-ID  U16   MAG TAPE UTILITY'
*B   'CVASEB'    '  DECK-ID  U17   MAG TAPE UTILITY'
*B   'ALCBUF'    '  DECK-ID  U18   MAG TAPE UTILITY'
*B   'NXTLOC'    '  NEXT AVAILABLE LOCATION'
*T
*K, I8
*N, MTUPFL... B
*Z
*CTO        MAGNETIC TAPE UTILITIES INSTALLED
*Z
```

SKELETON DIRECTS PROCESSING AS SHOWN:

ALL INPUT IS FOR LOGICAL UNIT 6. IF THE INSTALLATION IS TO BE MADE FROM ANOTHER LOGICAL UNIT, CHANGE I VALUE AS APPROPRIATE.

} CONTROL STATEMENTS USED DURING LIBEDT PROCESSING

NOTE: EACH LINE ENDS WITH A CARRIAGE RETURN AND A LINE FEED. DECK IDENTIFICATIONS ARE OPTIONAL.

Figure 11-1. Skeleton for Adding Multiple Tape Utility Processor to System

The operator loads the first two records just written on the installation tape:

```
LOAD, 6
```

After loading the records, the computer replies:

```
ANY MORE INPUT, ENTER LU
```

This SKED internal condition is cleared by pressing carriage return. The computer replies:

```
NEXT
```

The operator now inserts all the other records in the skeleton of figure 11-1. Each input record is followed by a carriage return and a line feed.

```
INSERT, 2, 4
```

```
*B   'MTUP'
*P, F, , TAPUTL
.
.
.
*Z
```

} Remaining records from skeleton in figure 11-1

carriage return

The final carriage return terminates the loading command sequence. The comment device displays:

NEXT

The operator may now list the full skeleton by:

CATLOG

When the computer replies:

NEXT

the operator is ready to dump the skeleton onto the installation tape, using the same magnetic tape as before:

REW, 6

When the tape is rewound and the computer replies (NEXT), the operator writes the skeleton with:

DUMP, 6

The computer replies with:

NEXT

and the skeleton is now saved on the output device. The operator exits from the skeleton editor with:

EXIT

11.2 BUILDING MTUP INSTALLATION FILE

The skeleton is now used in conjunction with the magnetic tape utility processor binaries to create the magnetic tape utility processor installation file. The utility program LIBILD generates the file. In the following example, both the new skeleton and the magnetic tape utility processor binaries are on magnetic tape, both mounted on logical unit 16.

Assuming that the job processor is still in control of the computer, LIBILD is put in control by the operator entering:

*LIBILD

The Comment Device Displays:	The Operator Replies:
CONTROL LU =	Carriage return
DEFS LU =	Carriage return
INSTALL LU =	Six carriage returns
NEWLIB LU =	Carriage return
LIB 01 LU =	16 carriage returns
LIB 02 LU =	Carriage return
SKELETON LU =	16 carriage returns

Note that only a carriage return is entered as the response to the query CONTROL LU = . This is because the sequence control statements are read from the comment device. A carriage return is also the response to the query DEFS LU = , since the installation file is to be created according to the skeleton and not according to a definitions deck. The response to INSTALL LU = indicates that the installation file is to be written on logical unit 6. No new output library is to be created. Therefore, a carriage return is entered following the query NEWLIB LU =

Following the 16-carriage-return reply to the query SKELETON LU=, the binary programs are read from logical unit 16 and saved on mass storage. When all the binary programs have been read, the comment device displays:

LOAD SKEL/INSTAL, CR WHEN READY

After mounting the skeleton tape prepared by SKED (section 11. 1) on logical unit 16, mounting the installation tape on logical unit 6, and readying both of these units, the operator replies with a carriage return. LIBILD reads the skeleton and prepares a magnetic tape utility processor installation file by reading the skeleton records and binaries, and processing the binaries according to the instructions of the skeleton records.

When the installation file is complete, the comment device displays:

LIBRARY BUILD COMPLETE
TYPE *Z TO TERMINATE OR
TYPE *C TO CONTINUE WITH CURRENT
SKELETON AND/OR
OUTPUT LIBRARY LU'S

Since the installation file is now complete and residing on logical unit 6, the operator exits from LIBILD by replying:

*Z

The system returns to job processor control.

11.3 ENTERING MTUP INTO MSOS

The newly prepared magnetic tape utility processor installation file may now be used by LIBEDT utility to enter magnetic tape utility processor into MSOS.

Assuming that the job processor is still controlling the computer, the library editor is placed in control by the operator entering:

*LIBEDT

When the library editor is controlling the computer, the comment device displays:

```
LIB IN
```

The operator loads the installation file on logical unit 6 and readies that tape unit. He then causes the file to be read by entering:

```
*V, 6
```

After LIBEDT has processed the file, the comment device displays the CTO statement entered at the end of the skeleton:

```
MAGNETIC TAPE UTILITIES INSTALLED
```

The first *Z statement from the skeleton causes LIBEDT termination; the second *Z statement causes job processor termination. MSOS is now augmented by addition of the magnetic tape utilities.

12.1 NEW FEATURES

MSOS 5 is a multiprogramming operating system designed to support a variety of applications requiring dedicated system utilization, batch processing, and program check-out features in a real-time environment. In addition to those features that presently exist in MSOS 4, the following features are provided for the CYBER 18-20 computer.

- MSOS 5 provides auto-data transfer (ADT) for pseudo direct memory transfers of data to and from a device.
- The system provides access to data in memory beyond the 128K byte boundary.
- All additional general purpose registers are saved and restored on interrupt.
- MSOS 5 supports a real-time clock in auto-data transfer mode.
- The system supports up to eight mass storage disk drives, each capable of storing 50 million 8-bit bytes of formatted data.
- Up to two flexible disk drives are supported. These are formatted in either IBM 3740 format (128 bytes per sector) or the CDC 1700 Series rotating mass storage format (192 bytes per sector).
- MSOS 5 provides a stand-alone background text editor for manipulation of user program and data files.
- Continual support of operation on 1704/1714/1774/1784 computers is provided.
- The additional instruction repertoire supported by the CYBER 18-20 computer is included within the macro assembler.
- Peripheral drivers 1C under MSOS include new drivers for the following peripheral equipment:

  -1833-1/1833-3/1867-10/1867-20 Storage Module Drive

  -1833-5/1865-1/1865-2 Flexible Disk
- The system provides the ability to advance records or files and backspace records or files from the job processor.

12.2 DEFICIENCIES AND LIMITATIONS

There are no known deficiencies or limitations in the system.

12.3 PSR LEVEL

The release level of MSOS 5 is summary level 110. (Summary levels for RPG II Version 1.0, FORTRAN Version 3.3A/B, and Magnetic Tape Utility Processor Version 2.0 may be less than 110 since they are previously released products.)

UPDATING A SYSTEM BY INSTALLATION OF LIBILD BINARY UPDATE FILES

Periodically, MSOS and its associated products are updated by Control Data. At the time of an update, the user is sent a binary update file for MSOS and a binary update file for each associated product in the user's system. The user also receives a COSY release file and a COSY corrections file for each product in his system. The COSY files may be used to generate new system listings.

To update the system, a skeleton corresponding to the latest installation file must be obtained. This may be done by using the program SKED as shown in appendix N. The skeleton should be modified to change the *S system initializer control statements defining SYSLVL, SYSMON, SYSDAY, and SYSYER, which define the PSR level and system build date. If the skeleton is on cards, cards can be manually changed. Otherwise, SKED may be used to make these changes. The operator then uses LIBILD to create a new installation file containing the modules from all binary update files the user has received. This is done by presenting the binary update files as library input to LIBILD together with the modified skeleton to create a new installation file. The new installation file may be used to build an updated system as described in section 3.7, omitting steps 2 through 5.

A system initializer error message may appear, indicating memory space has been exceeded. This may be due to an increase in size in one or more updated main memory resident modules. In this case BGNMON should be decreased. ENDOV4 must also be decreased if BGNMON = ENDOV4 (e.g., the system has neither partitioned core nor unused area; see appendix L). The user must determine the amount, L, of new space needed.

The new values are then computed:

$$BGNMON' = BGNMON - L = n_1$$

$$ENDOV4' = ENDOV4 - L = n_2$$

To modify the values of BGNMON and ENDOV4, the skeleton records defining BGNMON and ENDOV4 must be modified. The new records have the form:

*S, BGNMON, n_1

*S, ENDOV4, n_2

Where: n_1 and n_2 are the new values.

These records are found near the beginning of the skeleton.

The current size of unprotected may be ascertained by dumping the contents of the locations $F6_{16}$ and $F7_{16}$ and computing the difference.

$V = F6_{16} - F7_{16}$ = number of words of unprotected.

Let V' equal new size of unprotected. Then:

$$V' = V - L - m$$

Where: L is the number of words added to main memory resident programs.
m is the number (if any) of additional words added to SYSDAT.

To generate updated listings for each product, follow the instructions for cases 1 and 2 below.

CASE 1: CARD READER IN SYSTEM

1. Obtain a punched card copy of each COSY correction card image. (If the COSY correction file is already on cards, omit this step.)

   If there is a card punch in the system, LIBEDT may be used to transfer the COSY correction tape to cards. If there is no card punch in the system, obtain a listing of the COSY corrections as follows:

   Mount the COSY corrections file on logical unit p and ready the device. The following dialog then takes place on the comment device:

Comment device	Remarks
*JOB	Operator requests the job processor
J	Job processor is ready
*LIBEDT	Operator requests the library editor
LIB IN	Library editor is ready
*T, p, A, 9, A, , 1	Operator requests transfer of data: p = input logical unit, A = ASCII mode, 9 = output logical unit, for the one (installation) file
*Z	Operator exits from LIBEDT

This generates a listing of the corrections.

Using this listing, the operator punches a card for each correction card image listed.

2. Insert the corrections from step 1 into the source decks for those modules that have corrections. If source decks are not available and the system has a card punch, source decks may be generated as follows:

   Let p = COSY input device logical unit

   Let q = card punch logical unit

   Mount the COSY release file on logical unit p and ready the device. Mount the COSY corrections file into the standard input device. Enter the following from the comment device:

Comment Device	Remarks
*JOB	Request for job processor
*K, I10	Input on logical unit 10
*CSY, Ip, Pq	Reassign COSY input to logical unit p, output to logical unit q
*COSY	Execute COSY
*Z	Exit from the job processor

   COSY then punches source decks in Hollerith format.

   If there is no card punch in the system, let q equal the tape logical unit so that Hollerith source deck images are to be written to logical unit q. Use the above procedure to write Hollerith source deck images to logical unit q. A listing of the Hollerith source may be obtained using LIBEDT. A source deck may be punched from this listing.

3. Punch any necessary control cards and use the macro assembler and/or FORTRAN to process the corrected modules, obtaining the desired listings.

CASE 2: NO CARD READER IN SYSTEM

Obtain updated Hollerith output on tape. This may be done as follows:

1. Let n, p, and q be magnetic tape logical units. (Logical unit q may be a pseudo tape or simulated magnetic tape.) Mount the COSY release file on logical unit n and ready the device. Mount the COSY corrections file on logical unit p and ready the device. Enter the following from the comment device:

Comment Device	Remarks
*JOB	Request the job processor
*CSY, In, Pq	Reassign COSY input to logical unit r, output to logical unit q
*K, Ip	Reassign standard input to logical unit p
*COSY	Execute COSY
*Z	Exit from the job processor

   This generates Hollerith source deck images on logical unit q with the tape on logical q rewound. The system responds:

   J

   Enter on the comment device:

Comment Device	Remarks
*K, Iq, P2	Reassign the input to COSY output logical unit
*ASSEM	Execute the macro assembler.

   Watch the listing as printed. Compare it with the COSY index received with the COSY tapes from Control Data. The index indicates which programs are FORTRAN programs.

   At the end of the first block of assembly programs, the system automatically reverts to the job processor and outputs:

   J

   Enter:

   *FTN

   to execute FORTRAN.

   At the end of this block of FORTRAN programs, the system again automatically reverts to the job processor and outputs:

   J

   At this point enter:

   ASSEM

   Similarly alternate between FORTRAN and macro assembler as necessary until all desired listings have been obtained.

   An error message may appear indicating mass storage has been exceeded. This is because the default macro assembly options include load-and-go output to mass memory. This error message may be ignored.

GLOSSARY

The glossary is intended to assist in the communication of facts and ideas related to information processing.

In all instances, a comparison has been made to the American National Standards Institute (ANSI) glossary to ensure consistency with standard nomenclature wherever possible.

ALLOCATABLE MAIN MEMORY—That portion of main memory that can be assigned to programs by the core allocator (i.e. SYSDAT and resident program areas cannot be allocated). See appendix L.

AUTOLOAD—To place the resident routines of the operating system in main memory

BGNMON—Beginning address of the monitor

BINARY—A program (module) in binary format

BOOTSTRAP—A set of machine language instructions designed to read in a program from an input device and begin execution of that program

BYTE—A sequence of adjacent binary digits operated upon as a unit and usually shorter than a word; within the CYBER 18/1700 Series computer systems, a byte is eight bits; i.e., a byte is one-half of a 16-bit word

COSY—A format for compressing information in source decks or source deck images by replacing three or more sequential blanks with two special ASCII characters

CREP—Core resident Entry Point Table. Holds entry points (linkage addresses) to protected programs executed in part 1 of core.

CREP 1—Core Resident Entry Point 1 Table. Holds entry points (linkage addresses) to protected programs executed in part 1 of core.

DEADSTART—CYBER 18-20 hardware logic that allows execution of panel mode instructions input from an external input device. These instructions may load a bootstrap into macro memory and initiate its execution.

ENDOV4—End of blank common (see appendix L)

FILE MANAGER—An MSOS product for managing records and files

FORTRAN—Formula translating system; a language primarily used to express scientific computer programs by arithmetic formulas

INITIALIZER—An MSOS program that initializes the system using an installation file

INSTALLATION FILE—The file of installation material used to install MSOS on a computer system

LIBEDT—The library editor program

LIBILD—The library building program

MACRO ASSEMBLER—The program that compiles source language into 1700 machine language statements (ASSEM)

MAIN MEMORY RESIDENT—A program (e.g., SYSDAT, monitor) that always resides in main memory

MASS STORAGE RESIDENT—A part of the system that resides on mass storage and that is brought into core when needed by the system. Many of these programs are either in the system library or the program library.

METHOD 1—The full system installation method for adding a product. See section 5.

METHOD 2—The special installation method for adding new products that do not require changes to SYSDAT. See section 5.

MTUP—The magnetic tape utility program

ORDINAL—A number that specifies the order of programs on the system library. Loosely used to refer to the program designated by this ordinal

PROGRAM LIBRARY—Library of background programs. These can be relocatable binary or absolute (program files).

RPG—The report generator program

SKED—The skeleton editor

SORT/MERGE — The sorting and merging program for file data manipulation

SYSDAT — The system data base that is resident at the beginning of main memory

SYSTEM LIBRARY — The library of foreground programs for MSOS

VERIFY — The verify program; part of the installation file

Bootstrap entries for 1832-4 Magnetic Tape Controller with seven-track magnetic tape are as follows:

0822G
6846G
9871G
0102G
0131G
1803G
0814G
D870G
6872G
686DG
6871G
09FEG
6864G
8000G
3000G
6862G
5801G
0B00G
C000G
0908G
5840G
C8FBG
095EG
E000G
8009G
0B06G
0A01G
8000G
0900G
5837G
0A01G
8000G
0900G
5833G
CC58G
582DG
582CG
D855G
CC54G
0FC2G
0FE4G
4C52G
D851G
0FC2G
5824G
D84DG
CC4CG
5821G
0FC2G
0FE4G
4C49G
D848G
C846G
983BG
0122G
D843G
18E9G
C83FG
011BG
C83BG
8837G
683CG
8835G
6839G
5808G
C835G
6837G
8830G
6834G
5803G
1400G
0000G
0000G
E82AG
0DFEG
CE2EG
6E2CG
0141G
18FBG
1CF8G
0000G
0FC2G
0FE6G
1CFCG
0000G
E820G
0D08G
0B04G
0B00G
0B00G
0DF7G
0B05G
0A03G
6817G
C000G
0FFFG
09FFG
0101G
18FDG
E812G
0B04G

```
A000G
0002G
0101G
18F3G
C80BG
0102G
09FEG
18F0G
0B05G
1CE5G
8480G
1FFFG
3FFFG
0000G
1000G
0000G
0480G
0000G
0000G
0000G
0000G
0000G
0000G
```

Bootstrap entries for 1832-4 Magnetic Tape Controller with nine-track magnetic tape are as follows:

```
6819G
09FEG
6834G
8000G
2000G
6832G
5801G
0B00G
C000G
0108G
5810G
C8FBG
092EG
E000G
8009G
0B06G
0A01G
8000G
0100G
5807G
0A01G
8000G
0100G
5803G
1400G
0000G
0000G
E81FG
0D08G
0B04G
0B00G
0B00G
0DF7G
0B05G
0A03G
6816G
C000G
0FFFG
09FEG
0101G
18FDG
E811G
0B04G
A000G
0002G
0101G
18F3G
C80AG
0102G
09FEG
18F0G
0B05G
1CE5G
8480G
1FFFG
3FFFG
0000G
0000G
0480G
```

Bootstrap entries for the card reader are those in appendix C, excluding the first three lines and the final five lines.

The following is a listing of the deadstart deck, including a bootstrap to read from the card reader.

```
K71008000G
K0000G
L0500G
6823G
6823G
E000G
0581G
C000G
0080G
03FEG
0AD7G
681AG
0DFEG
0B00G
02FBG
A815G
0FC8G
6C16G
0B00G
02FEG
A810G
BC12G
6C11G
D810G
0829G
D89CG
C80BG
0121G
18F1G
C806G
086CG
0841G
0111G
1C05G
18E2G
0F0QG
00FFG
0000G
0000G
0000G
K0000G
J14G
K5000G
J10G
K31200800
```

A deadstart deck containing a bootstrap to read from a magnetic tape unit consists of the following three parts. The first symbol on each card must be in column one. There must be one blank between each pair of characters.

1. Initial cards:

```
K71008000G
K0000G
L
```

2. Cards containing the symbols are listed in appendix B for the 1832-4 Magnetic Tape Controller with either the seven-track or nine-track tape, depending on the type of installation device. These symbols may be grouped; e.g., five lines per card, if desired.

3. Final cards:

```
K0000G
J14G
K2400G
J10G
K31202800
```

D.1 1700 COMPUTER SYSTEMS

1784 COMPUTER

Loading

1. Press the STOP button.
2. Press the master CLEAR button on the console.
3. Set the row of switches near the bottom right hand side of the console to their neutral position. Set the MODE switch to 32K if the system size contains 32K of memory or less, or to 65K if the size is larger than 32K. The INSTRUCTION/CYCLE switch should be set to INSTRUCTION. All other two-position switches should be in the off position.
4. Select the P register by pressing the button marked P.
5. Set the pushbutton register to the first address in core that information is to be entered into. Do this by first pressing the CLEAR button to the right of the pushbutton register to clear the register. Then press the pushbuttons in the pattern that gives the hexadecimal address desired. (The starting address of the system initializer bootstraps is 0000.)
6. Set the ENTER/SWEEP switch to ENTER.
7. Select the X register.
8. Enter the code into memory as follows:
   a. Press the CLEAR button to the right of the pushbutton register.
   b. Enter the first (or next) word of code into the pushbutton register.
   c. Press the GO button.
   d. Repeat these steps for every word of code to be entered.
9. When finished, set the ENTER/SWEEP and the INSTRUCTION/CYCLE switches to the neutral position.

Checking

1. Press the master CLEAR button on the console.
2. Set the row of switches to the same positions as in step 3 above.
3. Select the P register.
4. Set the pushbutton register to the first address to be checked by first pressing the CLEAR button to the right of the pushbutton register and then pressing the pushbuttons that set the address in the register.
5. Set the ENTER/SWEEP switch to SWEEP.
6. Select the X register.
7. Press the GO button.
8. The data that is stored at the core address specified in step 4 appears in the pushbutton register. To display the next sequential words of core, press the GO button.

To check the address of any location during this procedure, select the P register and the core address appears in the pushbutton register. To resume checking the code, select the X register and continue pressing the GO button. When finished, set the ENTER/SWEEP switch and the INSTRUCTION/CYCLE switch to the neutral position.

1704, 1714, 1774 COMPUTERS

Loading

1. Put the RUN/STEP switch momentarily in the STEP position.
2. Press the master CLEAR switch.
3. All other switches should be set in the neutral or off position.
4. If there is a MODE switch (1714 computer), it should be set to 32K or 65K as required.
5. Select the P register.
6. Set the pushbutton register to the first address in core that information is to be entered into. Do this by first pressing the CLEAR button to the right of the pushbutton register and then setting the pushbuttons in the pattern that give the hexadecimal address desired (the starting address of the system initializer bootstraps is 0000).
7. Set the ENTER/SWEEP switch to ENTER.
8. Select the X register.

9. Enter the code into memory as follows:

   a. Press the CLEAR button to the right of the pushbutton register.

   b. Enter the first (or next) word of code into the pushbutton register.

   c. Momentarily put the RUN/STEP switch in the STEP position.

   d. Repeat these steps for every word of code to be entered.

10. When finished, set the ENTER/SWEEP switch to the neutral position.

Checking

1. Press the master CLEAR switch.
2. Set the row of switches to the same positions as in steps 3 and 4 under Loading above (1704, 1714, 1774 Computers).
3. Select the P register.
4. Set the pushbutton register to the first address to be checked as in step 6 under Loading above (1704, 1714, 1774 Computers).
5. Set the ENTER/SWEEP switch to SWEEP.
6. Select the X register.
7. Momentarily set the RUN/STEP switch to the STEP position.
8. The data stored at the core address specified in step 4 appears in the pushbutton register. To display the next and subsequent sequential words of core, momentarily set the RUN/STEP switch to the STEP position.

To check the address of any location during this procedure, select the P register and the core address appears in the pushbutton register. To resume checking the code, select the X register and continue pressing the RUN/STEP switch to the STEP position. When finished, set the ENTER/SWEEP switch and the INSTRUCTION/CYCLE switch to the neutral position.

D.2 CYBER 18-20 COMPUTER SYSTEM

The methods of loading a bootstrap are given in section 3.5 (cards) or 3.6 (magnetic tape). To check a bootstrap proceed as follows (this procedure assumes panel mode has been entered).

1. Enter:

   J11G

   Selects the P register

2. Enter:

   KhhhhG

   Sets P to the first address to be checked (for system initializer bootstrap, hhhh=0000)

3. Enter:

   J07G

   Selects macro memory (if not already selected)

4. Enter:

   LG

   Begins checking the bootstrap

5. Enter:

   G

   Displays the contents of the next location

Repeat step 5 until the bootstrap has been completely checked.

SYSTEM INITIALIZER CODES

The following defines the system initializer error codes:

Message	Significance
ERROR 1	Asterisk initiator missing
ERROR 2	Number appears in the name field
ERROR 3	Illegal control statement
ERROR 4	Input mode illegal
ERROR 5	Statement other than *Y or *YM previously entered
ERROR 6	Statement other than *Y previously entered
ERROR 7	*Y not entered prior to the first *L
ERROR 8	Name appears in the number field
ERROR 9	Illegal hexadecimal core relocation field
ERROR A	Illegal mass storage sector number
ERROR B	Error return from the loader module
ERROR C	Not used
ERROR D	Not used
ERROR E	Field terminator invalid
ERROR F	More than 120 characters in the control statement
ERROR 10	Ordinal name without ordinal number
ERROR 11	Doubly defined entry point
ERROR 12	Invalid ordinal number
ERROR 13	Loader control statement out of order — Correct order is L, LP, M, MP.
ERROR 14	Data declared during an *M load but not by the first segment; initialization restarted.
ERROR 15	Not used
ERROR 16	Irrecoverable mass storage input/output error
ERROR 17	Irrecoverable loader error; last program loaded was ignored.
ERROR 18	Not used
ERROR 19	Not used
ERROR 20	*S, END0V4, hhhh not defined before first *L
ERROR 21	*S, MSIZV4, hhhh not defined before first *LP or *MP
ERROR 22	Attempt to load part 1 core resident into nonexistent memory
ERROR 23	The name used in the second field of an *M control statement was not previously defined as an entry point.
ERROR 24	The entry point, SECTOR, was not defined at the start of initialization and is not available to the initializer.
ERROR 25	Illegal partition number in the first field of an *MP statement or illegal number of partitions in the second field of statement
ERROR 26	An attempt was made to load an *MP program when no partitioned core table exists in SYSDAT.

SYSTEM INITIALIZER LOADER ERRORS

Error	Significance
LOADER ERROR 1	Unrecognizable input
LOADER ERROR 2	Mass storage overflow
LOADER ERROR 3	Out-of-order input block
LOADER ERROR 4	Illegal data or common declaration
LOADER ERROR 5	Core overflow
LOADER ERROR 6	Overflow of entry point table
LOADER ERROR 7	Data block overflow
LOADER ERROR 8	Duplicate entry point
LOADER ERROR 9	15-/16-bit arithmetic error
LOADER ERROR 10	Unpatched externals
LOADER ERROR 11	Insufficient core for both SYSDAT and paging
LOADER ERROR 12	Illegal page number used
LOADER ERROR 13	Undefined transfer address
LOADER ERROR 14	Invalid function for loader
LOADER ERROR 15	Link table overflow
LOADER ERROR 16	External table overflow
LOADER ERROR 17	Entry point absolutized to $7FFF_{16}$

SYSTEM INITIALIZER DISK ERRORS

Error	Significance
DISK ERROR	Address tag write sequence attempted but internal/external reject found
DISK FAILURE xx	Surface test operation caused error xx. Refer to the device error codes to interpret xx.
DISK COMPARE ERROR SECT aaaa WORD bbbb IS cccc SB dddd	Surface test pattern error on sector aaaa at word bbbb. Only one error is listed per sector. Data read was cccc but it should be dddd.

1. Press STOP. Press MASTER CLEAR.
2. Press the mass memory AUTOLOAD button.
3. If the console has a MODE switch, set it to 32K or 65K, depending on the size of the system.

   NOTE

   When using a 1733-2 Cartridge Disk Controller, press the MASTER CLEAR button before going to step 4.

4. Activate GO or RUN.
5. The system outputs:

   MSOS 5.0 - - PSR LEVEL xx date

   Where: xx is the version number of the system.
   date is the date of system release.
6. If the PROGRAM PROTECT switch has not been set, the system outputs:

   SET PROGRAM PROTECT.

   If using a 1700 Series computer system, set the protect switch up.

   If using a CYBER 18-20 computer, press ESCAPE and enter:

   J28@

   This sets program protect and reverts to operator mode.
7. The system then outputs the name of the system (a parameter in SYSDAT).
8. The system outputs:

   32K MODE

   or

   65K MODE
9. If the system contains a file manager, it outputs:

   CHECKING FILES -

   If the files are found to be valid, the message OK is output. If errors are found, the user is given the option to continue or to purge all system files.
10. The system outputs:

    ENTER DATE/TIME MMDDYYHHMM
11. Enter the date and time in the form:

    mmddyyhhmm

    These items are (left to right, two digits each): month, day, year, hour (out of 24), minutes.
12. The system then outputs the date and time:

    DATE: dd month yy

    TIME: hh mm:00

The following procedures are required to initialize a disk pack for use under MSOS on an 1867-10 or 1867-20 Storage Module Drive.

1. Format the pack. This initializes the pack with the proper head gaps and sync patterns. It destroys any address tag information or data that may be on the pack.
2. Write address tags. This sets up the sector information for each sector on the pack.
3. Write data. MSOS requires that data initially be written on the entire pack. A disk error occurs if an attempt is made to read data from a sector that has never had data written in it.

G.1 PROCEDURES FOR FORMATTING A PACK (1867-10/20)

G.1.1 FORMATTING A PACK (1867-10/20 DISK) WITH A WORKING MSOS

1. Enter the job processor.
2. Enter on the comment device:

   *SMDMPI

3. The output on the comment device appears as follows:

   BOOTSTRAP INITIALIZER FIRST WORD ADDRESS WILL BE 2E90 MASTER CLEAR AND START AT THE ADDRESS ABOVE WITH
   A = DRIVE LOGICAL NUMBER
   Q = EQUIPMENT CODE (0XX0) OR ZERO IF EQUIP 14 (STANDARD)

4. Master clear the computer, mount the pack to be formatted, and ready the drive.
5. Follow the instructions on the comment device.
6. Watch the controller lights to see when formatting is finished; i.e., when lights stop flashing, the procedure requires approximately two minutes. On completion of the formatting operation, both the A and Q registers are zero if there was no error.

G.1.2 FORMATTING A PACK WITHOUT A WORKING MSOS

A formatting deadstart deck is supplied to the user along with the installation materials. This deck is not to be confused with the system initializer deadstart deck. This deck is used in the following procedure:

1. Mount the pack and ready the drive.
2. Press master clear.
3. Place the formatting deadstart deck in the card reader.
4. Push the RESET button on card reader to ready it.
5. Push the DEADSTART button.
6. The bootstrap within the deadstart deck is read into macro memory and begins execution automatically.
7. Proceed to step 6, section G.1.1 above.

G.2 PROCEDURES FOR WRITING ADDRESS TAGS AND DATA ON A PACK

G.2.1 WRITING ADDRESS TAGS AND DATA ON 1867-10/20 WITH A WORKING MSOS

1. Enter the job processor.
2. Enter on the comment device:

   *SILP
   carriage return

3. The message to turn off the protect switch is received.
4. Press ESCAPE and enter:

   J20@
   carriage return

   The message to enter the date is received.

5. Mount the disk pack to be initialized on the drive (unit 0) and make ready.
6. Enter the date in the form:

   mm/dd/yy.

The system responds with:

Q

7. Enter:

*0,4
carriage return

The system responds with:

Q

8. Enter:

*G
carriage return

9. The system outputs:

ENABLE ADDRESS WRITE -- THEN CR

10. Press carriage return.

11. Writing of address tags and data occurs. This procedure requires about 10 minutes for a single density pack, about 20 minutes for a double density pack. At the conclusion, the system outputs a Q.

G.2.2 WRITING ADDRESS TAGS AND DATA ON 1867-10/20 WITHOUT A WORKING MSOS

Use the *G function of the system initializer during system build.

An end-of-file card (figure H-1) contains only a 6, 7, 8, 9 punch in column 1.

The card following an end-of-file is the first card of the next file.

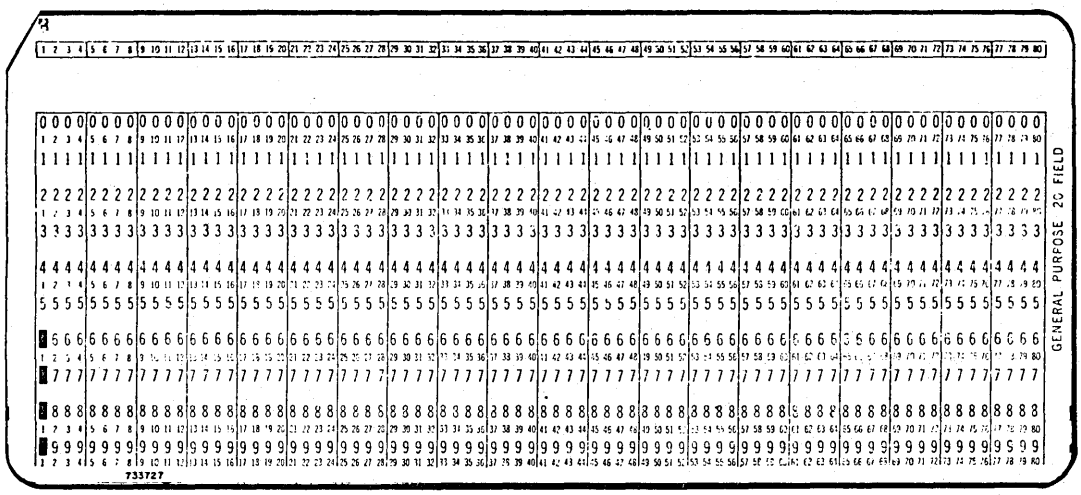


Figure H-1. End-of-File Card

```
  DATE: 09/24/76
*V
*S,SYSMON,$3039
*S,SYSDAY,$3233
*S,SYSYER,$3736
*S,SYSLVL,$3130
*V
*V        1700 MASS STORAGE OPERATING SYSTEM - VER 5.0
*V
*V        COPYRIGHT CONTROL DATA CORPORATION - 1976
*V
*V        MSOS 5.0 TEST SYSTEM 5
*V
*YM,LIBEDT,1
*YM,LOADSD,2
*YM,JOBENT,3
*YM,JOBPRO,4
*YM,PROTEC,5
*YM,JPLOAD,6
*YM,JPCMGE,7
*YM,JPT13,8
*YM,JCPOV4,9
*YM,JLGOV4,10
*YM,JPSTV4,11
*YM,NAMEV4,12
*YM,JPFLV4,13
*YM,APICV4,14
*YM,RESTOR,15
*YM,RCOVER,16
*YM,RMKPT,17
*YM,ODEBUG,18
*YM,SYSCOP,19
*YM,SYSSEG,20
*YM,MIPRO,21
*YM,TOFUNC,22
*YM,PSTOP,23
*YM,FFLIST,24
*YM,VERIFY,25
*YM,DUMMY1,26
*YM,DUMMY2,27
*YM,DUMMY3,28
*YM,DUMMY4,29
*YM,DUMMY5,30
*YM,DUMMY6,31
*YM,DUMMY7,32
*YM,DUMMY8,33
*YM,DUMMY9,34
*YM,DUMMY0,35
*S,N4,$0800
*S,ENDOM,$7FFF
*S,BGNMON,$863F
*S,MSIZV4,$FFFE
*S,SECTOR,$7FFE
*
*L        SYSTEM DATA PROGRAM
C57F7F
    SYSDAT     0000      MSOS 5.0 TEST SYSTEM 5              SUMMARY-104
*L        SPACE REQUEST PROCESSOR
    SPACE      18D7       DECK-ID M29  MSOS 5.0              SUMMARY-110
*
*         SYSTEM CORE RESIDENT PROGRAMS
*
*LP       MONITOR
    NMONI      883F       DECK-ID M10  MSOS 5.0              SUMMARY-110
    MOISP      8882       DECK-ID 058  MSOS 5.0              SUMMARY-110
    RW         8A30       DECK-ID M09  MSOS 5.0              SUMMARY-110
    T14        8ADE       DECK-ID M26  MSOS 5.0              SUMMARY-110
    T16        8AEF       DECK-ID M04  MSOS 5.0              SUMMARY-110
    PARAME     8AFA       DECK-ID M03  MSOS 5.0              SUMMARY-110
    COMMON     8B6B       DECK-ID 055  MSOS 5.0              SUMMARY-110
    NIPROC     8B9F       DECK-ID M12  MSOS 5.0              SUMMARY-110
    ALVOL      8C31       DECK-ID M16  MSOS 5.0              SUMMARY-110
    OFVOL      8C4E       DECK-ID M15  MSOS 5.0              SUMMARY-110
    ALCORE     8C5B       DECK-ID M17  MSOS 5.0              SUMMARY-110
    DCORE      8D09       DECK-ID 057  MSOS 5.0              SUMMARY-110
    PRTCOR     8E67       DECK-ID 056  MSOS 5.0              SUMMARY-110
    NFNR       C075       DECK-ID M21  MSOS 5.0              SUMMARY-110
    NCMPRQ     C0EA       DECK-ID M20  MSOS 5.0              SUMMARY-110
    MAKQ       C11A       DECK-ID M08  MSOS 5.0              SUMMARY-110
    ADEV       C148       DECK-ID M22  MSOS 5.0              SUMMARY-110
    TMINT      C2B8       DECK-ID M06  MSOS 5.0              SUMMARY-110
    DTIMER     C34D       DECK-ID M05  MSOS 5.0              SUMMARY-110
```

```
    TOD         C36F      DECK-ID M25  MSOS 5.0                    SUMMARY-110
    MINT        C38E      DECK-ID M07  MSOS 5.0                    SUMMARY-110
    TRVEC       C4CA      DECK-ID M14  MSOS 5.0                    SUMMARY-110
*LP       DEBUGGING / CHECKOUT
    SNAPOL      C51C      DECK-ID M02  MSOS 5.0                    SUMMARY-110
    ECMDMP      C5D5      DECK-ID D10  PERIPH. DRIVERS 1.1C        SUMMARY-110
    816331      C77E      DECK-ID C89  PERIPH. DRIVERS 1.1C        SUMMARY-110
*LP       FILE MANAGER
    FILMGR      C801      DECK-ID F01  FILE MANAGER                SUMMARY-110
    HSPCV4      CA79      DECK-ID F02  FILE MANAGER                SUMMARY-110
    SPHFIS      CH6A      DECK-ID F03  FILE MANAGER                SUMMARY-110
*LP       CORE RESIDENT DRIVERS
    FFDATA      CD3C      DECK-ID M27  MSOS 5.0                    SUMMARY-110
    DUMMY       CEA9      DECK-ID M30  MSOS 5.0                    SUMMARY-110
    ALAQ        CECC      DECK-ID M28  MSOS 5.0                    SUMMARY-110
    D18ECM      CF3R      DECK-ID C96  PERIPH. DRIVERS 1.1C        081276A-110
    D1810       CFE0      DECK-ID C25  PERIPH. DRIVERS 1.0C        SUMMARY-106
    DSMD        D1R7      DECK-ID C71  PERIPH. DRIVERS 1.1C        SUMMARY-110
    CSMD        D216      DECK-ID C72  PERIPH. DRIVERS 1.1C        SUMMARY-110
    ESMD        D28B      DECK-ID C73  PERIPH. DRIVERS 1.1C        SUMMARY-110
    HSMD        D2CF      DECK-ID C74  PERIPH. DRIVERS 1.1C        SUMMARY-110
    ALMERR      D2F8      DECK-ID C75  PERIPH. DRIVERS 1.1C        SUMMARY-110
    XSMD        D353      DECK-ID C76  PERIPH. DRIVERS 1.1C        SUMMARY-110
    SSMD        D477      DECK-ID C77  PERIPH. DRIVERS 1.1C        SUMMARY-110
    LIKDUM      D556      DECK-ID C78  PERIPH. DRIVERS 1.1C        SUMMARY-110
    ECCALG      D607      DECK-ID C79  PERIPH. DRIVERS 1.1C        SUMMARY-110
    NODIAS      D7AD      DECK-ID C87  PERIPH. DRIVERS 1.1C        SUMMARY-110
    DMYCP       D7B2      DECK-ID C83  PERIPH. DRIVERS 1.1C        SUMMARY-110
    SMD1DA      D7B8      DECK-ID C84  PERIPH. DRIVERS 1.1C        SUMMARY-110
    SMD1DR      D7BB      DECK-ID C85  PERIPH. DRIVERS 1.1C        SUMMARY-110
    DPSDSK      D7C1      DECK-ID C90  PERIPH. DRIVERS 1.1C        SUMMARY-110
    MMEXEC      D7F0      DECK-ID M01  MSOS 5.0                    SUMMARY-110
*LP       REENTRANT FORTRAN RUNTIME LIBRARY
    FORTR       D97F      DECK-ID A01  FTN 3.3 RUNTIME             SUMMARY-102
    OPRMR       DAC3      DECK-ID B01  FTN 3.3 RUNTIME             SUMMARY-102
    PARAMR      DAED      DECK-ID B02  FTN 3.3 RUNTIME             SUMMARY-102
    OPF2IR      DAFE      DECK-ID B03  FTN 3.3 RUNTIME             SUMMARY-102
    ARSR        DBAA      DECK-ID B04  FTN 3.3 RUNTIME             SUMMARY-102
    SQRTFR      DBC3      DECK-ID B05  FTN 3.3 RUNTIME             SUMMARY-102
    SIGNR       DC1E      DECK-ID B06  FTN 3.3 RUNTIME             SUMMARY-102
    FXFLTR      DC4A      DECK-ID B07  FTN 3.3 RUNTIME             SUMMARY-102
    EXPR        DC82      DECK-ID B08  FTN 3.3 RUNTIME             SUMMARY-102
    ALOGR       DD22      DECK-ID B09  FTN 3.3 RUNTIME             SUMMARY-102
    TANHR       DD99      DECK-ID B10  FTN 3.3 RUNTIME             SUMMARY-102
    SNCSR       DE05      DECK-ID B11  FTN 3.3 RUNTIME             SUMMARY-102
    ATANR       DECF      DECK-ID B12  FTN 3.3 RUNTIME             SUMMARY-102
    QBQIOR      DF6D      DECK-ID C01  FTN 3.3 RUNTIME             SUMMARY-102
    BINARR      E0D2      DECK-ID C02  FTN 3.3 RUNTIME             SUMMARY-102
    IOCODR      E11D      DECK-ID D01  FTN 3.3 RUNTIME             SUMMARY-102
    INITLR      E151      DECK-ID D02  FTN 3.3 RUNTIME             SUMMARY-102
    RSTORR      E160      DECK-ID D03  FTN 3.3 RUNTIME             SUMMARY-102
    GETCHR      E170      DECK-ID D04  FTN 3.3 RUNTIME             SUMMARY-102
    IPACKR      E18A      DECK-ID D05  FTN 3.3 RUNTIME             SUMMARY-102
    UPDATR      E1C8      DECK-ID D06  FTN 3.3 RUNTIME             SUMMARY-102
    DECPLR      E1D5      DECK-ID D07  FTN 3.3 RUNTIME             SUMMARY-102
    INTGRR      E1FB      DECK-ID D08  FTN 3.3 RUNTIME             SUMMARY-102
    SPACER      E228      DECK-ID D09  FTN 3.3 RUNTIME             SUMMARY-102
    HOLR        E240      DECK-ID D10  FTN 3.3 RUNTIME             SUMMARY-102
    DCMXR       E2D4      DECK-ID D11  FTN 3.3 RUNTIME             SUMMARY-102
    HXASCR      E347      DECK-ID D12  FTN 3.3 RUNTIME             SUMMARY-102
    AFMTOR      E39A      DECK-ID D13  FTN 3.3 RUNTIME             SUMMARY-102
    PFMTOR      E3C4      DECK-ID D14  FTN 3.3 RUNTIME             SUMMARY-102
    AFMTIR      E3DD      DECK-ID D15  FTN 3.3 RUNTIME             SUMMARY-102
    PFMTIP      E40B      DECK-ID D16  FTN 3.3 RUNTIME             SUMMARY-102
    ASCHXR      E422      DECK-ID D17  FTN 3.3 RUNTIME             SUMMARY-102
    HXDCR       E45D      DECK-ID D18  FTN 3.3 RUNTIME             SUMMARY-102
    FLOTIR      E4ED      DECK-ID D19  FTN 3.3 RUNTIME             SUMMARY-102
    FOUTR       E537      DECK-ID D20  FTN 3.3 RUNTIME             SUMMARY-102
    EOUTR       E5C1      DECK-ID D21  FTN 3.3 RUNTIME             SUMMARY-102
    EWRITR      E6A9      DECK-ID D22  FTN 3.3 RUNTIME             SUMMARY-102
    INTIIR      E6B5      DECK-ID D23  FTN 3.3 RUNTIME             SUMMARY-102
    FORMTR      E6D2      DECK-ID D24  FTN 3.3 RUNTIME             SUMMARY-102
    QBOFIR      E8A8      DECK-ID D25  FTN 3.3 RUNTIME             SUMMARY-102
    QBQFLR      E8C5      DECK-ID D26  FTN 3.3 RUNTIME             SUMMARY-102
    ORQFXR      E8F4      DECK-ID D27  FTN 3.3 RUNTIME             SUMMARY-102
    HEXAR       E928      DECK-ID D28  FTN 3.3 RUNTIME             SUMMARY-102
    HFXDR       E943      DECK-ID D29  FTN 3.3 RUNTIME             SUMMARY-102
    ASCIIR      E960      DECK-ID D30  FTN 3.3 RUNTIME             SUMMARY-102
    DECHXR      E975      DECK-ID D31  FTN 3.3 RUNTIME             SUMMARY-102
    AFORMR      E995      DECK-ID D32  FTN 3.3 RUNTIME             SUMMARY-102
    PFORMR      E9B1      DECK-ID D33  FTN 3.3 RUNTIME             SUMMARY-102
    FLOTGR      E9CD      DECK-ID D34  FTN 3.3 RUNTIME             SUMMARY-102
    FLOTR       E9E9      DECK-ID B14  FTN 3.3 RUNTIME             SUMMARY-102
    COMFPR      EC36      DECK-ID B15  FTN 3.3 RUNTIME             SUMMARY-102
    SGDBLR      ED49      DECK-ID E01  FTN 3.3 RUNTIME             SUMMARY-102
    QBD2IR      ED5D      DECK-ID E02  FTN 3.3 RUNTIME             SUMMARY-102
    DABSR       EDEF      DECK-ID E03  FTN 3.3 RUNTIME             SUMMARY-102
    DSQRTR      EE0B      DECK-ID E04  FTN 3.3 RUNTIME             SUMMARY-102
    DSIGNR      EE8B      DECK-ID E05  FTN 3.3 RUNTIME             SUMMARY-102
    DEXPR       EEB8      DECK-ID E08  FTN 3.3 RUNTIME             SUMMARY-102
    DLOGR       EF74      DECK-ID E09  FTN 3.3 RUNTIME             SUMMARY-102
```

```
    LSNCSR    F01A      DECK-ID E11  FTN 3.3 RUNTIME          SUMMARY-102
    DATANR    F12F      DECK-ID E12  FTN 3.3 RUNTIME          SUMMARY-102
    OPNDFR    F206      DECK-ID E14  FTN 3.3 RUNTIME          SUMMARY-102
    DOUTR     F222      DECK-ID E15  FTN 3.3 RUNTIME          SUMMARY-102
    DFLOTR    F32C      DECK-ID E13  FTN 3.3 RUNTIME          SUMMARY-102
    DRSTRR    F78A      DECK-ID E10  FTN 3.3 RUNTIME          SUMMARY-102
    NXTLOC    F7C0      NEXT AVAILABLE LOCATION
*LP,SF800
    ECM2RB    F800      DECK-ID D09  PERIPH. DRIVERS 1.1C     SUMMARY-110
*
*        SYSTEM MASS RESIDENT PROGRAMS
*
*M              LIBEDT      1
    LIBEDT    02B0      DECK-ID M35  MSOS 5.0                 SUMMARY-110
*M              LOADSD      2
    LOAD1     02EF      DECK-ID M36  MSOS 5.0                 SUMMARY-110
    BRNCH1    0336      DECK-ID M37  MSOS 5.0                 SUMMARY-110
    LIDRV1    0394      DECK-ID M38  MSOS 5.0                 SUMMARY-110
    LCDRV1    03E5      DECK-ID M39  MSOS 5.0                 SUMMARY-110
    LMDRV1    0412      DECK-ID M40  MSOS 5.0                 SUMMARY-110
    LLDRV1    0431      DECK-ID M41  MSOS 5.0                 SUMMARY-110
    ADJOF1    043F      DECK-ID M42  MSOS 5.0                 SUMMARY-110
    CNVRT1    044B      DECK-ID M43  MSOS 5.0                 SUMMARY-110
    LSTOT1    0463      DECK-ID M44  MSOS 5.0                 SUMMARY-110
    LINK11    0486      DECK-ID M45  MSOS 5.0                 SUMMARY-110
    LOADR1    04F9      DECK-ID M46  MSOS 5.0                 SUMMARY-110
    NAMPR1    0560      DECK-ID M47  MSOS 5.0                 SUMMARY-110
    RRDRZ1    060E      DECK-ID M48  MSOS 5.0                 SUMMARY-110
    ENTEX1    0707      DECK-ID M49  MSOS 5.0                 SUMMARY-110
    XFRPR1    0730      DECK-ID M50  MSOS 5.0                 SUMMARY-110
    STHASE    074E      DECK-ID M51  MSOS 5.0                 SUMMARY-110
    LNKENT    0836      DECK-ID M52  MSOS 5.0                 SUMMARY-110
    LNKCR1    0852      DECK-ID M53  MSOS 5.0                 SUMMARY-110
    PATCH     0893      DECK-ID M54  MSOS 5.0                 SUMMARY-110
    TRSCH1    08D5      DECK-ID M55  MSOS 5.0                 SUMMARY-110
    HASH      091D      DECK-ID M56  MSOS 5.0                 SUMMARY-110
    TRSTR1    0935      DECK-ID M57  MSOS 5.0                 SUMMARY-110
    PAGE      096A      DECK-ID M58  MSOS 5.0                 SUMMARY-110
    PROGLD    0A7A      DECK-ID M59  MSOS 5.0                 SUMMARY-110
    SCAN1     0B7C      DECK-ID M60  MSOS 5.0                 SUMMARY-110
    CMPU1     0C42      DECK-ID M61  MSOS 5.0                 SUMMARY-110
    ADJOV2    0C4F      DECK-ID M62  MSOS 5.0                 SUMMARY-110
    ADMPR1    0C68      DECK-ID M63  MSOS 5.0                 SUMMARY-110
*M              JOBENT      3
    JOBENT    0312      DECK-ID M64  MSOS 5.0                 SUMMARY-110
    T11       00CC      DECK-ID M65  MSOS 5.0                 SUMMARY-110
    T7        0103      DECK-ID M66  MSOS 5.0                 SUMMARY-110
    T5        0225      DECK-ID M67  MSOS 5.0                 SUMMARY-110
    T3        027E      DECK-ID M68  MSOS 5.0                 SUMMARY-110
*S,N1,P
*M              JOBPRO      4
    JOBPRO    031A      DECK-ID M69  MSOS 5.0                 SUMMARY-110
    ONE       0224      DECK-ID M70  MSOS 5.0                 SUMMARY-110
    TWO       022D      DECK-ID M71  MSOS 5.0                 SUMMARY-110
    THREE     0230      DECK-ID M72  MSOS 5.0                 SUMMARY-110
*S,N2,P
*M              PROTEC      5
    EPROTK    0320      DECK-ID D61  MSOS 5.0                 SUMMARY-110
    JBKILL    056C      DECK-ID M75  MSOS 5.0                 SUMMARY-110
*M              JPLOAD      6
    JPLOAD    0330      DECK-ID M76  MSOS 5.0                 SUMMARY-110
*M              JPCHGE      7
    JPCHGE    0335      DECK-ID M77  MSOS 5.0                 SUMMARY-110
    ASCHEX    013E      DECK-ID M78  MSOS 5.0                 SUMMARY-110
*M              JPT13       8
    T13       033A      DECK-ID M79  MSOS 5.0                 SUMMARY-110
*M              JCRDV4      9
    JCRDV4    033F      DECK-ID M80  MSOS 5.0                 SUMMARY-110
*M              JLGOV4     10
    JLGOV4    0343      DECK-ID M81  MSOS 5.0                 SUMMARY-110
*M              JPSTV4     11
    JPSTV4    0346      DECK-ID M84  MSOS 5.0                 SUMMARY-110
*M              NAMEV4     12
    NAMEV4    0348      DECK-ID M85  MSOS 5.0                 SUMMARY-110
*M              JPFLV4     13
    JPFLV4    0350      DECK-ID M82  MSOS 5.0                 SUMMARY-110
*M              JPFIV4     14
    JPF2V4    0357      DECK-ID M83  MSOS 5.0                 SUMMARY-110
*M              RESTOR     15
    RESTOR    0361      DECK-ID M86  MSOS 5.0                 SUMMARY-110
*M              RCOVER     16
    RCOVER    0364      DECK-ID M87  MSOS 5.0                 SUMMARY-110
    OUTSEL    0144      DECK-ID M88  MSOS 5.0                 SUMMARY-110
    RDMPV4    0169      DECK-ID M89  MSOS 5.0                 SUMMARY-110
    MASDMP    0249      DECK-ID M90  MSOS 5.0                 SUMMARY-110
*M              BRKPT      17
    BRKPT1    036D      DECK-ID N01  MSOS 5.0                 SUMMARY-110
*M              ODEBUG     18
    ODBUG1    037A      DECK-ID N07  MSOS 5.0                 SUMMARY-110
    GETREQ    0120      DECK-ID N08  MSOS 5.0                 SUMMARY-110
    LMAREQ    0240      DECK-ID N09  MSOS 5.0                 SUMMARY-110
```

```
    DPCREQ    0300      DECK-ID N10  MSOS 5.0          SUMMARY-110
    SCNREQ    03C0      DECK-ID N11  MSOS 5.0          SUMMARY-110
    SETREQ    0480      DECK-ID N12  MSOS 5.0          SUMMARY-110
    MBCREQ    04E0      DECK-ID N13  MSOS 5.0          SUMMARY-110
    SCHREQ    05A0      DECK-ID N14  MSOS 5.0          SUMMARY-110
    SPFREQ    0660      DECK-ID N15  MSOS 5.0          SUMMARY-110
    CPPREQ    06C0      DECK-ID N16  MSOS 5.0          SUMMARY-110
    SPPREQ    0720      DECK-ID N17  MSOS 5.0          SUMMARY-110
    ALMREQ    0780      DECK-ID N18  MSOS 5.0          SUMMARY-110
    SBMREQ    07E0      DECK-ID N19  MSOS 5.0          SUMMARY-110
    ALCREQ    0840      DECK-ID N20  MSOS 5.0          SUMMARY-110
    RELREQ    0900      DECK-ID N21  MSOS 5.0          SUMMARY-110
    DACREQ    0960      DECK-ID N22  MSOS 5.0          SUMMARY-110
    PTMREQ    0A80      DECK-ID N23  MSOS 5.0          SUMMARY-110
    MTMREQ    0AA0      DECK-ID N24  MSOS 5.0          SUMMARY-110
    MSOREQ    0C00      DECK-ID N25  MSOS 5.0          SUMMARY-110
    CLOREQ    0CC0      DECK-ID N26  MSOS 5.0          SUMMARY-110
    WCOREQ    0D20      DECK-ID N27  MSOS 5.0          SUMMARY-110
    LASREQ    0DA0      DECK-ID N28  MSOS 5.0          SUMMARY-110
    DASREQ    0E40      DECK-ID N29  MSOS 5.0          SUMMARY-110
    MLUREQ    0F60      DECK-ID N30  MSOS 5.0          SUMMARY-110
    DPTREQ    0FC0      DECK-ID N31  MSOS 5.0          SUMMARY-110
    SLOREQ    1080      DECK-ID N32  MSOS 5.0          SUMMARY-110
    CWAREQ    10E0      DECK-ID N33  MSOS 5.0          SUMMARY-110
    DMMREQ    1200      DECK-ID N34  MSOS 5.0          SUMMARY-110
    SMMREQ    12C0      DECK-ID N35  MSOS 5.0          SUMMARY-110
    SMPREQ    1440      DECK-ID N36  MSOS 5.0          SUMMARY-110
    LSPREQ    1500      DECK-ID N37  MSOS 5.0          SUMMARY-110
    DSPREQ    15C0      DECK-ID N38  MSOS 5.0          SUMMARY-110
    DWSREQ    16E0      DECK-ID N39  MSOS 5.0          SUMMARY-110
    LSOREQ    1860      DECK-ID N40  MSOS 5.0          SUMMARY-110
    CCCREQ    1980      DECK-ID N41  MSOS 5.0          SUMMARY-110
    CCMREQ    1AA0      DECK-ID N42  MSOS 5.0          SUMMARY-110
    CMMREQ    1BC0      DECK-ID N43  MSOS 5.0          SUMMARY-110
    MMMREQ    1CE0      DECK-ID N44  MSOS 5.0          SUMMARY-110
    LICREQ    1E60      DECK-ID N45  MSOS 5.0          SUMMARY-110
    LIOREQ    1FE0      DECK-ID N46  MSOS 5.0          SUMMARY-110
    LAMREQ    2160      DECK-ID N47  MSOS 5.0          SUMMARY-110
    DUPREQ    22E0      DECK-ID N48  MSOS 5.0          SUMMARY-110
    LDPREQ    23A0      DECK-ID N49  MSOS 5.0          SUMMARY-110
    LDOREQ    24C0      DECK-ID N50  MSOS 5.0          SUMMARY-110
    DMOREQ    2640      DECK-ID N51  MSOS 5.0          SUMMARY-110
    WIKREQ    27C0      DECK-ID N52  MSOS 5.0          SUMMARY-110
    LSTREQ    28E0      DECK-ID N53  MSOS 5.0          SUMMARY-110
    PPINT     2A60      DECK-ID N54  MSOS 5.0          SUMMARY-110
    GETFLD    2B20      DECK-ID N55  MSOS 5.0          SUMMARY-110
    ASHX      2BE0      DECK-ID N56  MSOS 5.0          SUMMARY-110
    DMPBUF    2C40      DECK-ID N57  MSOS 5.0          SUMMARY-110
    ASCDEC    2D00      DECK-ID N58  MSOS 5.0          SUMMARY-110
    HXAS      2D60      DECK-ID N59  MSOS 5.0          SUMMARY-110
    DECOMP    2DC0      DECK-ID N60  MSOS 5.0          SUMMARY-110
    PETMM     2E80      DECK-ID N61  MSOS 5.0          SUMMARY-110
    PNTMD     2FA0      DECK-ID N62  MSOS 5.0          SUMMARY-110
    MASOT     3060      DECK-ID N63  MSOS 5.0          SUMMARY-110
    CONFM     3180      DECK-ID N64  MSOS 5.0          SUMMARY-110
    GETINT    32A0      DECK-ID N65  MSOS 5.0          SUMMARY-110
    FLCVSG    3360      DECK-ID N66  MSOS 5.0          SUMMARY-110
    FLCVDB    3480      DECK-ID N67  MSOS 5.0          SUMMARY-110
    NAMEMS    35A0      DECK-ID N68  MSOS 5.0          SUMMARY-110
    DCONV     3A20      DECK-ID N69  MSOS 5.0          SUMMARY-110
    LAZY2     3B40      DECK-ID N70  MSOS 5.0          SUMMARY-110
    ODDFLT    3C00      DECK-ID N71  MSOS 5.0          SUMMARY-110
    ODDFTN    3CC0      DECK-ID N72  MSOS 5.0          SUMMARY-110
    ECONV     4080      DECK-ID N73  MSOS 5.0          SUMMARY-110
    LAZY1     4140      DECK-ID N74  MSOS 5.0          SUMMARY-110
    ODFLOT    4200      DECK-ID N75  MSOS 5.0          SUMMARY-110
    ODFXFL    4440      DECK-ID N76  MSOS 5.0          SUMMARY-110
*M              SYSCOP      19
    SYSCOP    0432      DECK-ID N77  MSOS 5.0          SUMMARY-110
*M              SYSSEG      20
    CO1ST     0437      DECK-ID N78  MSOS 5.0          SUMMARY-110
    CO2ND     04E0      DECK-ID N79  MSOS 5.0          SUMMARY-110
    CO3RD     0D80      DECK-ID N80  MSOS 5.0          SUMMARY-110
    COLAST    1500      DECK-ID N81  MSOS 5.0          SUMMARY-110
*M              MIPRO       21
    MIPRO     0470      DECK-ID 062  MSOS 5.0          SUMMARY-110
     * * *  U N P A T C H E D    E X T E R N A L S    * * *
  CRIMPT
  INDACS
  TSUTIL
*M              TDFUNC      22
    TDFUNC    0477      DECK-ID 067  MSOS 5.0          SUMMARY-110
*M              EFSTOR      23
    EFSTOR    0476      DECK-ID N04  MSOS 5.0          SUMMARY-110
*M              EFLIST      24
    EFLIST    0480      DECK-ID N05  MSOS 5.0          SUMMARY-110
*M              VERIFY      25
    VERFY1    049D      DECK-ID 024  MSOS 5.0          SUMMARY-110
*M              DUMMY1      26
*M              DUMMY2      27
```

```
*M              DUMMY3      28
*M              DUMMY4      29
*M              DUMMY5      30
*M              DUMMY6      31
*M              DUMMY7      32
*M              DUMMY8      33
*M              DUMMY9      34
*M              DUMMY0      35
*
*       MASS RESIDENT DRIVERS
*
*M                1832-5 CASSETTE TAPE
    D18325    04A2      DECK-ID C33  PERIPH. DRIVERS 1.1C  SUMMARY-110
    K18325    004E      DECK-ID C34  PERIPH. DRIVERS 1.1C  SUMMARY-110
    IOCAS     00E9      DECK-ID C35  PERIPH. DRIVERS 1.1C  SUMMARY-110
    FS2CAS    00F8      DECK-ID C36  PERIPH. DRIVERS 1.1C  SUMMARY-110
    MCAS      0113      DECK-ID C37  PERIPH. DRIVERS 1.1C  SUMMARY-110
    NEXCAS    0159      DECK-ID C38  PERIPH. DRIVERS 1.1C  SUMMARY-110
    RECCAS    0170      DECK-ID C39  PERIPH. DRIVERS 1.1C  SUMMARY-110
    XCAS      01EC      DECK-ID C40  PERIPH. DRIVERS 1.1C  SUMMARY-110
    VCAS      022B      DECK-ID C41  PERIPH. DRIVERS 1.1C  SUMMARY-110
    WAITCS    0258      DECK-ID C42  PERIPH. DRIVERS 1.1C  SUMMARY-110
    NXTLOC    026E      NEXT AVAILABLE LOCATION
*S,S18325,S
*S,L18325,P
*M                COSY DRIVER
    DCOSY     04A9      DECK-ID M34  MSOS 5.0              SUMMARY-110
    NXTLOC    02F1      NEXT AVAILABLE LOCATION
*S,SCOSY,S
*S,LCOSY,P
*M                1860-72/92 MAG TAPE
    D1860     04B1      DECK-ID C13  PERIPH. DRIVERS 1.0C  SUMMARY-106
    K1860     0072      DECK-ID C14  PERIPH. DRIVERS 1.0C  SUMMARY-106
    CKREQ     0125      DECK-ID C15  PERIPH. DRIVERS 1.0C  SUMMARY-106
    REWCKL    015E      DECK-ID C16  PERIPH. DRIVERS 1.0C  SUMMARY-106
    FORMIT    0192      DECK-ID C17  PERIPH. DRIVERS 1.0C  SUMMARY-106
    WAIT      01CD      DECK-ID C18  PERIPH. DRIVERS 1.0C  SUMMARY-106
    XMOT      01DC      DECK-ID C19  PERIPH. DRIVERS 1.0C  SUMMARY-106
    XFER      0234      DECK-ID C20  PERIPH. DRIVERS 1.0C  SUMMARY-106
    SEATIO    0290      DECK-ID C21  PERIPH. DRIVERS 1.0C  SUMMARY-106
    RECVRY    0312      DECK-ID C24  PERIPH. DRIVERS 1.0C  SUMMARY-106
    TK7DAT    047F      DECK-ID C22  PERIPH. DRIVERS 1.0C  SUMMARY-106
    TK7       05F9      DECK-ID C23  PERIPH. DRIVERS 1.0C  SUMMARY-106
    NXTLOC    0612      NEXT AVAILABLE LOCATION
*S,S1860,S
*S,L1860,P
*M                PSEUDO TAPE
    DPSUDO    04C2      DECK-ID F04  FILE MANAGER          SUMMARY-110
    NXTLOC    03CF      NEXT AVAILABLE LOCATION
*S,SPSUDO,S
*S,LPSUDO,P
*M                1827-30/60 LINE PRINTER
    D1827     04CD      DECK-ID C01  PERIPH. DRIVERS 1.0C  SUMMARY-106
    K1827     002E      DECK-ID C02  PERIPH. DRIVERS 1.0C  SUMMARY-106
    LPWAIT    0111      DECK-ID C03  PERIPH. DRIVERS 1.0C  SUMMARY-106
    CKGINT    0115      DECK-ID C04  PERIPH. DRIVERS 1.0C  SUMMARY-106
    EDIT      0125      DECK-ID C05  PERIPH. DRIVERS 1.0C  SUMMARY-106
    DUMMY     016E      DECK-ID C06  PERIPH. DRIVERS 1.0C  SUMMARY-106
    NXTLOC    018C      NEXT AVAILABLE LOCATION
*S,S1827,S
*S,L1827,P
*M                1829-3/6 CARD READER
    D1829     0402      DECK-ID C07  PERIPH. DRIVERS 1.0C  SUMMARY-106
    K1829     002D      DECK-ID C08  PERIPH. DRIVERS 1.0C  SUMMARY-106
    FORMT     00D3      DECK-ID C09  PERIPH. DRIVERS 1.0C  SUMMARY-106
    FAULTN    0266      DECK-ID C10  PERIPH. DRIVERS 1.0C  SUMMARY-106
    ESTAT     028A      DECK-ID C11  PERIPH. DRIVERS 1.0C  SUMMARY-106
    CKPRMS    0293      DECK-ID C12  PERIPH. DRIVERS 1.0C  SUMMARY-106
    CM026     02A5      DECK-ID C26  PERIPH. DRIVERS 1.0C  SUMMARY-106
    NXTLOC    02C6      NEXT AVAILABLE LOCATION
*S,S1829,S
*S,L1829,P
*
*       MASS RESIDENT FILE MANAGER
*
*M
    DEFFIL    040A      DECK-ID F05  FILE MANAGER          SUMMARY-110
    FILSPC    0183      DECK-ID F06  FILE MANAGER          SUMMARY-110
    APEND     0230      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP01,S
*M
    RELFIL    04E0      DECK-ID F08  FILE MANAGER          SUMMARY-110
    RELSPC    0092      DECK-ID F09  FILE MANAGER          SUMMARY-110
    APEND     0141      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP02,S
*M
    DEFIDX    04E4      DECK-ID F10  FILE MANAGER          SUMMARY-110
    SORTFM    0098      DECK-ID F11  FILE MANAGER          SUMMARY-110
    FILSPC    0048      DECK-ID F06  FILE MANAGER          SUMMARY-110
    APEND     0155      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP03,S
```

```
*M
    LOKFIL    04E8      DECK-ID F12  FILE MANAGER          SUMMARY-110
    RPEND     004D      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP04,S
*M
    UPLFIL    04E9      DECK-ID F13  FILE MANAGER          SUMMARY-110
    RPEND     0030      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP05,S
*M
    STOSEQ    04EA      DECK-ID F14  FILE MANAGER          SUMMARY-110
    FILSPC    00C6      DECK-ID F06  FILE MANAGER          SUMMARY-110
    RPEND     0173      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP06,S
*M
    STODIR    04EE      DECK-ID F15  FILE MANAGER          SUMMARY-110
    RPEND     008D      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP07,S
*M
    STOIDX    04F0      DECK-ID F20  FILE MANAGER          SUMMARY-110
    HASHCD    0341      DECK-ID F16  FILE MANAGER          SUMMARY-110
    GETKID    034C      DECK-ID F17  FILE MANAGER          SUMMARY-110
    FILSPC    0373      DECK-ID F06  FILE MANAGER          SUMMARY-110
    RPEND     0420      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP08,S
*M
    RTVSEQ    04F8      DECK-ID F18  FILE MANAGER          SUMMARY-110
    RTNSPC    018A      DECK-ID F19  FILE MANAGER          SUMMARY-110
    RPEND     0226      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP09,S
*M
    RTVDIR    0501      DECK-ID F21  FILE MANAGER          SUMMARY-110
    RTNSPC    0115      DECK-ID F19  FILE MANAGER          SUMMARY-110
    RPEND     0185      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP10,S
*M
    RTVIDX    0506      DECK-ID F22  FILE MANAGER          SUMMARY-110
    HASHCD    025A      DECK-ID F16  FILE MANAGER          SUMMARY-110
    GETKID    0263      DECK-ID F17  FILE MANAGER          SUMMARY-110
    RTNSPC    02BA      DECK-ID F19  FILE MANAGER          SUMMARY-110
    RPEND     032A      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP11,S
*M
    RTVIDO    050F      DECK-ID F23  FILE MANAGER          SUMMARY-110
    GETKID    0260      DECK-ID F17  FILE MANAGER          SUMMARY-110
    RTNSPC    0287      DECK-ID F19  FILE MANAGER          SUMMARY-110
    RPEND     0327      DECK-ID F07  FILE MANAGER          SUMMARY-110
*S,FMRP12,S
*M
    FMDUMY    0518      DECK-ID F24  FILE MANAGER          SUMMARY-110
*S,FMREND,S
*S,MEGFMS,S     SPECIFY THE SYSTEM FILE SPACE
*M,MEGFMS+$0688
*M
    FMDUMY    10D0      DECK-ID F24  FILE MANAGER          SUMMARY-110
*S,JFILV4,S     SPECIFY THE JOB FILE TABLE SPACE
*M,JFILV4+2
*I       END OF SYSTEM
   * * *  U N P A T C H E D   E X T E R N A L S   * * *
  PARITY
  POWERU
```

```
JOB,INSTAL,SYSTEM
 1700 MASS STORAGE OPERATING SYSTEM VERSION 5.0    DATE OF RUN: 09/24/76   SYSTEM ID:  MSOS 5.0 TEST SYSTEM 5  (09/23/76)

IIIIIIIIIIII     NNN       NNN     SSSSSSSSSS     TTTTTTTTTTTT     AAAAAAAAAA     LLL
IIIIIIIIIIII     NNN       NNN    SSSSSSSSSSSS    TTTTTTTTTTTT    AAAAAAAAAAAA    LLL
IIIIIIIIIIII     NNN       NNN    SSSSSSSSSSSS    TTTTTTTTTTTT    AAAAAAAAAAAA    LLL
     III         NNNN      NNN    SSS      SSS        TTT         AAA      AAA    LLL
     III         NNNNN     NNN    SSS                 TTT         AAA      AAA    LLL
     III         NNNNNN    NNN    SSS                 TTT         AAA      AAA    LLL
     III         NNN NNN   NNN     SSSSSSSSSS         TTT         AAAAAAAAAAAA    LLL
     III         NNN  NNN  NNN    SSSSSSSSSSS         TTT         AAAAAAAAAAAA    LLL
     III         NNN   NNN NNN    SSSSSSSSSS          TTT         AAAAAAAAAAAA    LLL
     III         NNN    NNNNNN          SSS           TTT         AAA      AAA    LLL
     III         NNN     NNNNN          SSS           TTT         AAA      AAA    LLL
     III         NNN      NNNN   SSS     SSS          TTT         AAA      AAA    LLL
IIIIIIIIIIII     NNN       NNN    SSSSSSSSSSS        TTT         AAA      AAA    LLLLLLLLLLLL
IIIIIIIIIIII     NNN       NNN    SSSSSSSSSSS        TTT         AAA      AAA    LLLLLLLLLLLL
IIIIIIIIIIII     NNN       NNN     SSSSSSSSSS        TTT         AAA      AAA    LLLLLLLLLLLL

*K,I6
*LIBEDT
 LIB

 IN

 *K,I6
 IN

 *V  DEFINE REQUEST PRIORITIES
 IN

 *S,001,03,M
 IN

 *S,002,00,M
 IN

 *S,003,01,M
 IN

 *S,004,02,M
 IN

 *S,005,03,M
 IN

 *S,006,02,M
 IN

 *S,007,02,M
 IN

 *S,008,02,M
 IN
```

```
*S,009,02,M
IN

*S,010,02,M
IN

*S,011,02,M
IN

*S,012,03,M
IN

*S,013,03,M
IN

*S,014,03,M
IN

*S,015,02,M
IN

*S,016,03,M
IN

*S,017,03,M
IN

*S,018,04,M
IN

*S,019,04,M
IN

*S,020,04,M
IN

*S,021,04,M
IN

*S,022,04,M
IN

*S,023,04,M
IN

*S,024,04,M
IN

*S,025,04,M
IN

*S,026,04,M
IN

*S,027,04,M
IN

*S,028,04,M
IN

*S,029,04,M
IN

*S,030,04,M
IN

*S,031,04,M
IN

*S,032,04,M
IN

*S,033,04,M
IN

*S,034,04,M
IN

*S,035,04,M
IN

*V  1700 MACRO ASSEMBLER 3.
IN

*K,I6
IN

*L,LIBMAC
IN
```

```
*L,ASSEM
IN

*K,P8
IN

*P,F
  PASS1   2DD3    DECK-ID A02  MACRO ASSEMBLER       SUMMARY-110
  PA1PR2  3985    DECK-ID A03  MACRO ASSEMBLER       SUMMARY-110
  NXTLOC  3E35    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,PASS1,,,8
IN

*K,I6
IN

*K,P8
IN

*P,F
  PASS2   2DD3    DECK-ID A04  MACRO ASSEMBLER       SUMMARY-110
  PA2PR2  32AC    DECK-ID A05  MACRO ASSEMBLER       SUMMARY-110
  NXTLOC  378C    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,PASS2,,,8
IN

*K,I6
IN

*K,P8
IN

*P,F
  PASS3   2DD3    DECK-ID A06  MACRO ASSEMBLER       SUMMARY-110
  PA3PR2  3276    DECK-ID A07  MACRO ASSEMBLER       SUMMARY-110
  PA3PR3  3683    DECK-ID A08  MACRO ASSEMBLER       SUMMARY-110
  NXTLOC  3H89    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,PASS3,,,8
IN

*K,I6
IN

*K,P8
IN

*P,F
  TABLST  2DD3    DECK-ID A09  MACRO ASSEMBLER       SUMMARY-110
  NXTLOC  3557    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,TABLST,,,8
IN

*K,I6
IN

*K,P8
IN

*P,F
  XREF    2DD3    DECK-ID A10  MACRO ASSEMBLER       SUMMARY-110
  NXTLOC  33E5    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,XREF,,,8
IN

*K,I6
IN
```

```
*N,MACSKL,,,B
IN

*N,MACROS,,,B
IN

*V  TEXT EDITOR
IN

*K,I6
IN

*L,EDITOR
IN

*K,P8
IN

*P
  EDITFL  20D3    DECK-ID F26  FILE MANAGER          SUMMARY-110
  NXTLOC  3C2E    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,EDITFL,,,B
IN

*V
IN

*V      SORT/MERGE 1.0
IN

*V
IN

*K,I6
IN

*L,SMC
IN

*K,P8
IN

*P
  SMCMON  20D3    DECK-ID S02  SMC 1.0              SUMMARY-108
  FLOTN   360C    DECK-ID S06  SMC 1.0              SUMMARY-108
  PARABN  3919    DECK-ID S07  SMC 1.0              SUMMARY-108
  COMNFP  3929    DECK-ID S09  SMC 1.0              SUMMARY-108
  NXTLOC  3A5C    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SMCMON,,,B
IN

*K,I6
IN

*P
  SMCEDT  20D3    DECK-ID S03  SMC 1.0              SUMMARY-108
  NXTLOC  347D    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SMCEDT,,,B
IN

*K,I6
IN

*P
  SMCSRT  20D3    DECK-ID S04  SMC 1.0              SUMMARY-108
  NXTLOC  3013    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SMCSRT,,,B
IN
```

```
*K,I6
IN

*P
  SMCIMG  2DD3    DECK-ID S05  SMC 1.0                SUMMARY-108
  NXTLOC  2FFF    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SMCIMG,,,8
IN

*K,I6
IN

*P
  SMCFMG  2DD3    DECK-ID S06  SMC 1.0                SUMMARY-108
  NXTLOC  301D    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SMCFMG,,,8
IN

*V  FTN 3.3B COMPILER
IN

*V
IN

*K,I6
IN

*K,P8
IN

*L,FTN
IN

*P
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOA     3545    DECK-ID 02F  FORTRAN 3.3B          SUMMARY-102
  PHASEA  35AB    DECK-ID 07A  FORTRAN 3.3B          SUMMARY-102
  IOPRBA  3AF6    DECK-ID 08F  FORTRAN 3.3B          SUMMARY-102
  OPRMS   3EA2    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CFIVOC  3EBC    DECK-ID 34A  FORTRAN 3.3B          SUMMARY-102
  CKNAME  3F1A    DECK-ID 36A  FORTRAN 3.3B          SUMMARY-102
  CNVT    3F2A    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  CONV    3F6B    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  DIAG    3F9B    DECK-ID 04F  FORTRAN 3.3B          SUMMARY-102
  DIAGRG  4047    DECK-ID 37F  FORTRAN 3.3B          SUMMARY-102
  DXPS    4063    DECK-ID 05F  FORTRAN 3.3B          SUMMARY-102
  DFLOT   418F    DECK-ID 06F  FORTRAN 3.3B          SUMMARY-102
  DUMVOL  43C6    DECK-ID 35F  FORTRAN 3.3B          SUMMARY-102
  GETC    43F5    DECK-ID 14F  FORTRAN 3.3B          SUMMARY-102
  GETF    4420    DECK-ID 04A  FORTRAN 3.3B          SUMMARY-102
  GETSYM  482D    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  GPUT    4866    DECK-ID 02A  FORTRAN 3.3B          SUMMARY-102
  IGETCF  486F    DECK-ID 15F  FORTRAN 3.3B          SUMMARY-102
  PACK    48AB    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  RDLABL  48CD    DECK-ID 10A  FORTRAN 3.3B          SUMMARY-102
  STORE   496B    DECK-ID 11F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL  49C1    DECK-ID 03A  FORTRAN 3.3B          SUMMARY-102
  ENDDO   457E    DECK-ID 29A  FORTRAN 3.3B          SUMMARY-102
  GNST    4C83    DECK-ID 05A  FORTRAN 3.3B          SUMMARY-107
  HEADER  51BB    DECK-ID 36F  FORTRAN 3.3B          SUMMARY-102
  OPTION  51FD    DECK-ID 16F  FORTRAN 3.3B          SUMMARY-102
  OUTENT  529C    DECK-ID 06A  FORTRAN 3.3B          SUMMARY-102
  PLABEL  5200    DECK-ID 08A  FORTRAN 3.3B          SUMMARY-102
  STCHAR  5326    DECK-ID 11A  FORTRAN 3.3B          SUMMARY-102
  TYPE    5356    DECK-ID 12A  FORTRAN 3.3B          SUMMARY-102
  SAVEID  5564    DECK-ID 13A  FORTRAN 3.3B          SUMMARY-102
  LOCLA1  565A    DECK-ID 12F  FORTRAN 3.3B          SUMMARY-102
  DUMYA1  5720    DECK-ID 13F  FORTRAN 3.3B          SUMMARY-109
  QBQBUS  5787    DECK-ID 09A  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  5787    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3A1,,,8
IN

*K,I6
IN
```

```
*P,,,MARKER
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOA     3545    DECK-ID 02F  FORTRAN 3.3B          SUMMARY-102
  PHASEA  35AB    DECK-ID 07A  FORTRAN 3.3B          SUMMARY-102
  IOPRBA  3AF6    DECK-ID 08F  FORTRAN 3.3B          SUMMARY-102
  OBPRMS  3EA2    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CFIVOC  3EBC    DECK-ID 34A  FORTRAN 3.3B          SUMMARY-102
  CKNAME  3F1A    DECK-ID 36A  FORTRAN 3.3B          SUMMARY-102
  CNVT    3F2A    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  CONV    3F68    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  DIAG    3F98    DECK-ID 04F  FORTRAN 3.3B          SUMMARY-102
  DIAGRG  4047    DECK-ID 37F  FORTRAN 3.3B          SUMMARY-102
  DXP9    4063    DECK-ID 05F  FORTRAN 3.3B          SUMMARY-102
  DFLOT   418F    DECK-ID 06F  FORTRAN 3.3B          SUMMARY-102
  DUMVOL  43C6    DECK-ID 35F  FORTRAN 3.3B          SUMMARY-102
  GETC    43F5    DECK-ID 14F  FORTRAN 3.3B          SUMMARY-102
  GETF    4420    DECK-ID 04A  FORTRAN 3.3B          SUMMARY-102
  GETSYM  482D    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  GPUT    4866    DECK-ID 02A  FORTRAN 3.3B          SUMMARY-102
  IGETCF  488F    DECK-ID 15F  FORTRAN 3.3B          SUMMARY-102
  PACK    48A8    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  RDLABL  48CD    DECK-ID 10A  FORTRAN 3.3B          SUMMARY-102
  STORE   4968    DECK-ID 11F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL  49C1    DECK-ID 03A  FORTRAN 3.3B          SUMMARY-102
  ENDDO   4B7E    DECK-ID 29A  FORTRAN 3.3B          SUMMARY-102
  GNST    4C83    DECK-ID 05A  FORTRAN 3.3B          SUMMARY-107
  HEADER  5188    DECK-ID 36F  FORTRAN 3.3B          SUMMARY-102
  OPTION  51FD    DECK-ID 16F  FORTRAN 3.3B          SUMMARY-102
  OUTENT  529C    DECK-ID 06A  FORTRAN 3.3B          SUMMARY-102
  PLABEL  52D0    DECK-ID 08A  FORTRAN 3.3B          SUMMARY-102
  STCHAR  5326    DECK-ID 11A  FORTRAN 3.3B          SUMMARY-102
  TYPE    5358    DECK-ID 12A  FORTRAN 3.3B          SUMMARY-102
  SAVEID  5584    DECK-ID 13A  FORTRAN 3.3B          SUMMARY-102
  LOCLA2  565A    DECK-ID 18F  FORTRAN 3.3B          SUMMARY-109
  DUMYA2  5720    DECK-ID 19F  FORTRAN 3.3B          SUMMARY-109
  RYEQPR  5787    DECK-ID 19A  FORTRAN 3.3B          SUMMARY-102
  CHECKF  5978    DECK-ID 20A  FORTRAN 3.3B          SUMMARY-102
  COMNPR  5A2E    DECK-ID 15A  FORTRAN 3.3B          SUMMARY-102
  CONSUB  5AC5    DECK-ID 30A  FORTRAN 3.3B          SUMMARY-102
  DATAPR  5B4C    DECK-ID 31A  FORTRAN 3.3B          SUMMARY-102
  DIMPR   6048    DECK-ID 16A  FORTRAN 3.3B          SUMMARY-102
  EXRLPR  61F2    DECK-ID 24A  FORTRAN 3.3B          SUMMARY-102
  FGETC   6250    DECK-ID 21A  FORTRAN 3.3B          SUMMARY-102
  FORK    6329    DECK-ID 22A  FORTRAN 3.3B          SUMMARY-102
  SUBPPR  6525    DECK-ID 23A  FORTRAN 3.3B          SUMMARY-102
  TYPEPR  65CE    DECK-ID 18A  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  65E5    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3A2,,,B
IN

*K,I6
IN

*P,,,MARKER
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOA     3545    DECK-ID 02F  FORTRAN 3.3B          SUMMARY-102
  PHASEA  35AB    DECK-ID 07A  FORTRAN 3.3B          SUMMARY-102
  IOPRBA  3AF6    DECK-ID 08F  FORTRAN 3.3B          SUMMARY-102
  OBPRMS  3EA2    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CFIVOC  3EBC    DECK-ID 34A  FORTRAN 3.3B          SUMMARY-102
  CKNAME  3F1A    DECK-ID 36A  FORTRAN 3.3B          SUMMARY-102
  CNVT    3F2A    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  CONV    3F68    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  DIAG    3F98    DECK-ID 04F  FORTRAN 3.3B          SUMMARY-102
  DIAGRG  4047    DECK-ID 37F  FORTRAN 3.3B          SUMMARY-102
  DXP9    4063    DECK-ID 05F  FORTRAN 3.3B          SUMMARY-102
  DFLOT   418F    DECK-ID 06F  FORTRAN 3.3B          SUMMARY-102
  DUMVOL  43C6    DECK-ID 35F  FORTRAN 3.3B          SUMMARY-102
  GETC    43F5    DECK-ID 14F  FORTRAN 3.3B          SUMMARY-102
  GETF    4420    DECK-ID 04A  FORTRAN 3.3B          SUMMARY-102
  GETSYM  482D    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  GPUT    4866    DECK-ID 02A  FORTRAN 3.3B          SUMMARY-102
  IGETCF  488F    DECK-ID 15F  FORTRAN 3.3B          SUMMARY-102
  PACK    48A8    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  RDLABL  48CD    DECK-ID 10A  FORTRAN 3.3B          SUMMARY-102
  STORE   4968    DECK-ID 11F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL  49C1    DECK-ID 03A  FORTRAN 3.3B          SUMMARY-102
  ENDDO   4B7E    DECK-ID 29A  FORTRAN 3.3B          SUMMARY-102
  GNST    4C83    DECK-ID 05A  FORTRAN 3.3B          SUMMARY-107
  HEADER  5188    DECK-ID 36F  FORTRAN 3.3B          SUMMARY-102
  OPTION  51FD    DECK-ID 16F  FORTRAN 3.3B          SUMMARY-102
  OUTENT  529C    DECK-ID 06A  FORTRAN 3.3B          SUMMARY-102
  PLABEL  52D0    DECK-ID 08A  FORTRAN 3.3B          SUMMARY-102
  STCHAR  5326    DECK-ID 11A  FORTRAN 3.3B          SUMMARY-102
  TYPE    5358    DECK-ID 12A  FORTRAN 3.3B          SUMMARY-102
```

```
  SAVEID  55B4    DECK-ID 13A  FORTRAN 3.3B          SUMMARY-102
  LOCLA3  565A    DECK-ID 20F  FORTRAN 3.3B          SUMMARY-102
  DUMYA3  5720    DECK-ID 21F  FORTRAN 3.3B          SUMMARY-109
  ARAYSZ  5787    DECK-ID 42A  FORTRAN 3.3B          SUMMARY-102
  ASEMPR  580A    DECK-ID 40A  FORTRAN 3.3B          SUMMARY-102
  ASGNPR  59C8    DECK-ID 32A  FORTRAN 3.3B          SUMMARY-102
  BUOPR   5A0E    DECK-ID 33A  FORTRAN 3.3B          SUMMARY-102
  CHECKF  5B4A    DECK-ID 20A  FORTRAN 3.3B          SUMMARY-102
  CKIVC   5BFD    DECK-ID 35A  FORTRAN 3.3B          SUMMARY-102
  CONSUB  5C0D    DECK-ID 30A  FORTRAN 3.3B          SUMMARY-102
  CPLOOP  5C94    DECK-ID 43A  FORTRAN 3.3B          SUMMARY-102
  FGETC   5D3A    DECK-ID 21A  FORTRAN 3.3B          SUMMARY-102
  FORK    5E13    DECK-ID 22A  FORTRAN 3.3B          SUMMARY-102
  ERRPR   600F    DECK-ID 38A  FORTRAN 3.3B          SUMMARY-102
  MODMXR  606D    DECK-ID 39A  FORTRAN 3.3B          SUMMARY-102
  PUNT    66F3    DECK-ID 27A  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  6728    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3A3,,,B
IN

*K,I6
IN

*P,,,MARKER
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOA     3545    DECK-ID 02F  FORTRAN 3.3B          SUMMARY-102
  PHASEA  35AB    DECK-ID 07A  FORTRAN 3.3B          SUMMARY-102
  IOPRBA  3AF6    DECK-ID 08F  FORTRAN 3.3B          SUMMARY-102
  OBPRMS  3EA2    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CFIVOC  3EBC    DECK-ID 34A  FORTRAN 3.3B          SUMMARY-102
  CKNAME  3F1A    DECK-ID 36A  FORTRAN 3.3B          SUMMARY-102
  CNVT    3F2A    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  CONV    3F68    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  DIAG    3F9B    DECK-ID 04F  FORTRAN 3.3B          SUMMARY-102
  DIAGRG  4047    DECK-ID 37F  FORTRAN 3.3B          SUMMARY-102
  DXP9    4063    DECK-ID 05F  FORTRAN 3.3B          SUMMARY-102
  DFLOT   418F    DECK-ID 06F  FORTRAN 3.3B          SUMMARY-102
  DUMVOL  43C6    DECK-ID 35F  FORTRAN 3.3B          SUMMARY-102
  GETC    43F5    DECK-ID 14F  FORTRAN 3.3B          SUMMARY-102
  GETF    4420    DECK-ID 04A  FORTRAN 3.3B          SUMMARY-102
  GETSYM  482D    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  GPUT    4866    DECK-ID 02A  FORTRAN 3.3B          SUMMARY-102
  IGETCF  488F    DECK-ID 15F  FORTRAN 3.3B          SUMMARY-102
  PACK    48A8    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  RDLABL  48CD    DECK-ID 10A  FORTRAN 3.3B          SUMMARY-102
  STORE   4968    DECK-ID 11F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL  49C1    DECK-ID 03A  FORTRAN 3.3B          SUMMARY-102
  ENDDO   4B7E    DECK-ID 29A  FORTRAN 3.3B          SUMMARY-102
  GAST    4C83    DECK-ID 05A  FORTRAN 3.3B          SUMMARY-107
  HEADER  51BB    DECK-ID 36F  FORTRAN 3.3B          SUMMARY-102
  OPTION  51FD    DECK-ID 16F  FORTRAN 3.3B          SUMMARY-102
  OUTENT  529C    DECK-ID 06A  FORTRAN 3.3B          SUMMARY-102
  PLABEL  5200    DECK-ID 08A  FORTRAN 3.3B          SUMMARY-102
  STCHAR  5326    DECK-ID 11A  FORTRAN 3.3B          SUMMARY-102
  TYPE    5358    DECK-ID 12A  FORTRAN 3.3B          SUMMARY-102
  SAVEID  55B4    DECK-ID 13A  FORTRAN 3.3B          SUMMARY-102
  LOCLA4  565A    DECK-ID 22F  FORTRAN 3.3B          SUMMARY-102
  DUMYA4  5720    DECK-ID 23F  FORTRAN 3.3B          SUMMARY-109
  ARITH   5787    DECK-ID 14A  FORTRAN 3.3B          SUMMARY-102
  SUBSCR  5E20    DECK-ID 17A  FORTRAN 3.3B          SUMMARY-102
  TREE    60F4    DECK-ID 41A  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  6611    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3A4,,,B
IN

*K,I6
IN

*P,,,MARKER
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOA     3545    DECK-ID 02F  FORTRAN 3.3B          SUMMARY-102
  PHASEA  35AB    DECK-ID 07A  FORTRAN 3.3B          SUMMARY-102
  IOPRBA  3AF6    DECK-ID 08F  FORTRAN 3.3B          SUMMARY-102
  OBPRMS  3EA2    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CFIVOC  3EBC    DECK-ID 34A  FORTRAN 3.3B          SUMMARY-102
  CKNAME  3F1A    DECK-ID 36A  FORTRAN 3.3B          SUMMARY-102
  CNVT    3F2A    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  CONV    3F68    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  DIAG    3F9B    DECK-ID 04F  FORTRAN 3.3B          SUMMARY-102
  DIAGRG  4047    DECK-ID 37F  FORTRAN 3.3B          SUMMARY-102
```

```
  EXP9     4063    DECK-ID 05F  FORTRAN 3.3B          SUMMARY-102
  PLOT     418F    DECK-ID 06F  FORTRAN 3.3B          SUMMARY-102
  DUMVOL   43C6    DECK-ID 35F  FORTRAN 3.3B          SUMMARY-102
  GETC     43F5    DECK-ID 14F  FORTRAN 3.3B          SUMMARY-102
  GETF     4420    DECK-ID 04A  FORTRAN 3.3B          SUMMARY-102
  GETSYM   482D    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  GPUT     4866    DECK-ID 02A  FORTRAN 3.3B          SUMMARY-102
  IGETCF   488F    DECK-ID 15F  FORTRAN 3.3B          SUMMARY-102
  PACK     48AA    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  PDLABL   48CD    DECK-ID 10A  FORTRAN 3.3B          SUMMARY-102
  STORE    4964    DECK-ID 11F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL   49C1    DECK-ID 03A  FORTRAN 3.3B          SUMMARY-102
  ENDDO    487E    DECK-ID 29A  FORTRAN 3.3B          SUMMARY-102
  GNST     4C83    DECK-ID 05A  FORTRAN 3.3B          SUMMARY-107
  HEADER   5186    DECK-ID 36F  FORTRAN 3.3B          SUMMARY-102
  OPTION   51FD    DECK-ID 16F  FORTRAN 3.3B          SUMMARY-102
  OUTENT   529C    DECK-ID 06A  FORTRAN 3.3B          SUMMARY-102
  PLABEL   52D0    DECK-ID 08A  FORTRAN 3.3B          SUMMARY-102
  STCHAR   5326    DECK-ID 11A  FORTRAN 3.3B          SUMMARY-102
  TYPE     5358    DECK-ID 12A  FORTRAN 3.3B          SUMMARY-102
  SAVEID   5584    DECK-ID 13A  FORTRAN 3.3B          SUMMARY-102
  LOCLAS   565A    DECK-ID 24F  FORTRAN 3.3B          SUMMARY-109
  DUMYAS   5720    DECK-ID 25F  FORTRAN 3.3B          SUMMARY-109
  BDOPR    5787    DECK-ID 33A  FORTRAN 3.3B          SUMMARY-102
  CKIVC    58C3    DECK-ID 35A  FORTRAN 3.3B          SUMMARY-102
  IOSPR    58D3    DECK-ID 37A  FORTRAN 3.3B          SUMMARY-102
  PEQVS    5F19    DECK-ID 25A  FORTRAN 3.3B          SUMMARY-102
  PRNTNM   634D    DECK-ID 26A  FORTRAN 3.3B          SUMMARY-102
  SYMSCN   63DC    DECK-ID 26A  FORTRAN 3.3B          SUMMARY-102
  ENDLOC   63F8    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3A5...B
IN

*K,I6
IN

*P
  FTN33B   2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GO4      3583    DECK-ID 26F  FORTRAN 3.3B          SUMMARY-102
  PHASER   35C9    DECK-ID 21B  FORTRAN 3.3B          SUMMARY-102
  IOPRBB   3A73    DECK-ID 27F  FORTRAN 3.3B          SUMMARY-102
  GAPRMS   3FE7    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  CNVT     4001    DECK-ID 01A  FORTRAN 3.3B          SUMMARY-102
  DUMMY    403F    DECK-ID 01B  FORTRAN 3.3B          SUMMARY-102
  FCMSTK   414E    DECK-ID 02B  FORTRAN 3.3B          SUMMARY-102
  GETSYM   4108    DECK-ID 07F  FORTRAN 3.3B          SUMMARY-102
  KCPART   4211    DECK-ID 03B  FORTRAN 3.3B          SUMMARY-102
  KOUTPT   4242    DECK-ID 04B  FORTRAN 3.3B          SUMMARY-102
  KPCSTK   4254    DECK-ID 05B  FORTRAN 3.3B          SUMMARY-102
  KPC3PR   480C    DECK-ID 06B  FORTRAN 3.3B          SUMMARY-102
  KSYMGN   4824    DECK-ID 07B  FORTRAN 3.3B          SUMMARY-102
  LABKPC   486C    DECK-ID 08B  FORTRAN 3.3B          SUMMARY-102
  LABLER   4880    DECK-ID 09B  FORTRAN 3.3B          SUMMARY-102
  PUNT     489E    DECK-ID 10B  FORTRAN 3.3B          SUMMARY-102
  CONV     48C1    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  STORER   48F4    DECK-ID 34F  FORTRAN 3.3B          SUMMARY-102
  SYMBOL   4928    DECK-ID 11B  FORTRAN 3.3B          SUMMARY-102
  TSALOC   49C5    DECK-ID 12B  FORTRAN 3.3B          SUMMARY-102
  ARAYSZ   4A86    DECK-ID 42A  FORTRAN 3.3B          SUMMARY-102
  ASSEM    4B07    DECK-ID 13B  FORTRAN 3.3B          SUMMARY-102
  BANANA   4B77    DECK-ID 14B  FORTRAN 3.3B          SUMMARY-102
  BGINDO   4C42    DECK-ID 15B  FORTRAN 3.3B          SUMMARY-102
  END      4D4B    DECK-ID 16B  FORTRAN 3.3B          SUMMARY-102
  ENTCOD   4D9C    DECK-ID 17B  FORTRAN 3.3B          SUMMARY-102
  HELEN    4E6C    DECK-ID 18B  FORTRAN 3.3B          SUMMARY-102
  INXRST   4FC3    DECK-ID 19B  FORTRAN 3.3B          SUMMARY-102
  NOPROC   4FD7    DECK-ID 20B  FORTRAN 3.3B          SUMMARY-102
  READIR   5014    DECK-ID 22B  FORTRAN 3.3B          SUMMARY-102
  SUBFUN   506C    DECK-ID 23B  FORTRAN 3.3B          SUMMARY-102
  SYMSCN   50D3    DECK-ID 28A  FORTRAN 3.3B          SUMMARY-102
  ACP      50EF    DECK-ID 24B  FORTRAN 3.3B          SUMMARY-102
  AFIDL    55C2    DECK-ID 25B  FORTRAN 3.3B          SUMMARY-102
  ASUPER   563A    DECK-ID 26B  FORTRAN 3.3B          SUMMARY-102
  CGOTO    56F0    DECK-ID 27B  FORTRAN 3.3B          SUMMARY-102
  FINK     5784    DECK-ID 28B  FORTRAN 3.3B          SUMMARY-102
  INTRAN   5840    DECK-ID 29B  FORTRAN 3.3B          SUMMARY-102
  PARTSB   5A55    DECK-ID 30B  FORTRAN 3.3B          SUMMARY-102
  SUBPR1   5B03    DECK-ID 31B  FORTRAN 3.3B          SUMMARY-102
  SUBPR2   5B41    DECK-ID 32B  FORTRAN 3.3B          SUMMARY-102
  SUBPR3   5C27    DECK-ID 33B  FORTRAN 3.3B          SUMMARY-102
  ARITHR   5C6E    DECK-ID 34B  FORTRAN 3.3B          SUMMARY-102
  ENDLOC   5E7E    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN
```

```
*N,FTN3H1,,,H
IN

*K,I6
IN

*P
  FTN33H  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOC     3AB5    DECK-ID 28F  FORTRAN 3.3B          SUMMARY-102
  PHASEC  3AA3    DECK-ID 13C  FORTRAN 3.3B          SUMMARY-102
  IOPRHC  3E18    DECK-ID 29F  FORTRAN 3.3B          SUMMARY-102
  Q8PRMS  4CB3    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  BKDWN   4CCD    DECK-ID 01C  FORTRAN 3.3B          SUMMARY-102
  RLOUP   402C    DECK-ID 02C  FORTRAN 3.3B          SUMMARY-102
  BSS     406F    DECK-ID 03C  FORTRAN 3.3B          SUMMARY-102
  CHK*D   4D8D    DECK-ID 04C  FORTRAN 3.3B          SUMMARY-102
  CHOP    4F18    DECK-ID 05C  FORTRAN 3.3B          SUMMARY-102
  CL12    51A2    DECK-ID 06C  FORTRAN 3.3B          SUMMARY-102
  CON     529F    DECK-ID 07C  FORTRAN 3.3B          SUMMARY-102
  COUNT   52F2    DECK-ID 08C  FORTRAN 3.3B          SUMMARY-102
  DATAST  5310    DECK-ID 09C  FORTRAN 3.3B          SUMMARY-102
  GETSYM  53F5    DECK-ID 10C  FORTRAN 3.3B          SUMMARY-102
  INOUT   5499    DECK-ID 11C  FORTRAN 3.3B          SUMMARY-102
  IXOPT   5508    DECK-ID 12C  FORTRAN 3.3B          SUMMARY-102
  LABEL   564E    DECK-ID 14C  FORTRAN 3.3B          SUMMARY-102
  LABIN   5670    DECK-ID 15C  FORTRAN 3.3B          SUMMARY-102
  QXLD    56D6    DECK-ID 16C  FORTRAN 3.3B          SUMMARY-102
  REED    576A    DECK-ID 17C  FORTRAN 3.3B          SUMMARY-102
  SKIP    57C7    DECK-ID 18C  FORTRAN 3.3B          SUMMARY-102
  SYMSCN  581D    DECK-ID 19C  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  5839    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3C1,,,B
IN

*K,I6
IN

*P
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOOD    3AB5    DECK-ID 30F  FORTRAN 3.3B          SUMMARY-102
  PHASE6  3AAC    DECK-ID 14D  FORTRAN 3.3B          SUMMARY-102
  IOPRBD  3B49    DECK-ID 31F  FORTRAN 3.3B          SUMMARY-102
  Q8PRMS  40E1    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  AMOUT   40FB    DECK-ID 01D  FORTRAN 3.3B          SUMMARY-102
  ADMAX   46BA    DECK-ID 02D  FORTRAN 3.3B          SUMMARY-102
  BEGINO  48C4    DECK-ID 03D  FORTRAN 3.3B          SUMMARY-102
  BKDWN   4A31    DECK-ID 04D  FORTRAN 3.3B          SUMMARY-102
  COUNT   4A9A    DECK-ID 05D  FORTRAN 3.3B          SUMMARY-102
  FINISH  4AB8    DECK-ID 06D  FORTRAN 3.3B          SUMMARY-102
  GETSYM  4C3C    DECK-ID 10C  FORTRAN 3.3B          SUMMARY-102
  IACON   4CE0    DECK-ID 07D  FORTRAN 3.3B          SUMMARY-102
  IHCON   4D3A    DECK-ID 08D  FORTRAN 3.3B          SUMMARY-102
  INDEX   4D67    DECK-ID 09D  FORTRAN 3.3B          SUMMARY-102
  LAROUT  4D83    DECK-ID 10D  FORTRAN 3.3B          SUMMARY-102
  NP2OUT  4E63    DECK-ID 11D  FORTRAN 3.3B          SUMMARY-102
  NPUNCH  4E92    DECK-ID 12D  FORTRAN 3.3B          SUMMARY-102
  NWRITE  4FDC    DECK-ID 13D  FORTRAN 3.3B          SUMMARY-102
  PACK    5010    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  REDX    5042    DECK-ID 15D  FORTRAN 3.3B          SUMMARY-102
  RAPK    507E    DECK-ID 16D  FORTRAN 3.3B          SUMMARY-102
  SYMSCN  5048    DECK-ID 17D  FORTRAN 3.3B          SUMMARY-102
  TABDEC  50C*    DECK-ID 18D  FORTRAN 3.3B          SUMMARY-102
  UNPUNC  5148    DECK-ID 19D  FORTRAN 3.3B          SUMMARY-102
  CONV    515E    DECK-ID 33F  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  5197    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3D1,,,B
IN

*K,I6
IN

*P
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOE     3AB5    DECK-ID 32F  FORTRAN 3.3B          SUMMARY-102
  PHASE6  3AAB    DECK-ID 14E  FORTRAN 3.3B          SUMMARY-102
  IOPRBD  3B48    DECK-ID 31F  FORTRAN 3.3B          SUMMARY-102
  Q6PRMS  40E0    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  AMOUT   40FA    DECK-ID 01E  FORTRAN 3.3B          SUMMARY-102
  ADMAX   46C9    DECK-ID 02E  FORTRAN 3.3B          SUMMARY-102
  BEGINO  48D3    DECK-ID 03E  FORTRAN 3.3B          SUMMARY-102
  BKDWN   4A75    DECK-ID 04E  FORTRAN 3.3B          SUMMARY-102
```

```
  CONV    4ADE    DECK-ID 33F  FORTRAN 3.3B          SUMMARY-102
  COUNT   4B17    DECK-ID 05E  FORTRAN 3.3B          SUMMARY-102
  FINISH  4B2E    DECK-ID 06E  FORTRAN 3.3B          SUMMARY-102
  GETSYM  4CH2    DECK-ID 10C  FORTRAN 3.3B          SUMMARY-102
  IACON   4D56    DECK-ID 07E  FORTRAN 3.3B          SUMMARY-102
  IHCON   4D80    DECK-ID 08E  FORTRAN 3.3B          SUMMARY-102
  INDEX   4DDC    DECK-ID 09E  FORTRAN 3.3B          SUMMARY-102
  LAHOUT  4DF8    DECK-ID 10E  FORTRAN 3.3B          SUMMARY-102
  NP2OUT  4F17    DECK-ID 11E  FORTRAN 3.3B          SUMMARY-102
  NPUNCH  4F4F    DECK-ID 12E  FORTRAN 3.3B          SUMMARY-102
  NWRITE  5099    DECK-ID 13E  FORTRAN 3.3B          SUMMARY-102
  PACK    50DA    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  RHDX    50FF    DECK-ID 15E  FORTRAN 3.3B          SUMMARY-102
  RHPK    513C    DECK-ID 16E  FORTRAN 3.3B          SUMMARY-102
  SETPRT  5166    DECK-ID 17E  FORTRAN 3.3B          SUMMARY-102
  SYMSCN  52EE    DECK-ID 17D  FORTRAN 3.3B          SUMMARY-102
  TABDEC  530A    DECK-ID 18E  FORTRAN 3.3B          SUMMARY-102
  UNPUNC  5386    DECK-ID 19E  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  539C    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3E1,,,B
IN

*K,I6
IN

*P
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  GOF     3A85    DECK-ID 38F  FORTRAN 3.3B          SUMMARY-102
  SYMSCN  3A8F    DECK-ID 28A  FORTRAN 3.3B          SUMMARY-102
  PHASEF  3AA8    DECK-ID 01G  FORTRAN 3.3B          SUMMARY-102
  QBPRMS  3D6C    DECK-ID 10F  FORTRAN 3.3B          SUMMARY-102
  GETSYM  3D86    DECK-ID 02G  FORTRAN 3.3B          SUMMARY-102
  ACON    3DE6    DECK-ID 03G  FORTRAN 3.3B          SUMMARY-102
  HCON    3E45    DECK-ID 04G  FORTRAN 3.3B          SUMMARY-102
  LWRITE  3E78    DECK-ID 05G  FORTRAN 3.3B          SUMMARY-102
  MATCH   3EFF    DECK-ID 06G  FORTRAN 3.3B          SUMMARY-102
  SORT    3F52    DECK-ID 07G  FORTRAN 3.3B          SUMMARY-102
  IPEPAK  3F87    DECK-ID 08G  FORTRAN 3.3B          SUMMARY-102
  CONV    400F    DECK-ID 03F  FORTRAN 3.3B          SUMMARY-102
  GETSYR  4042    DECK-ID 39F  FORTRAN 3.3B          SUMMARY-102
  TITLE   4057    DECK-ID 40F  FORTRAN 3.3B          SUMMARY-102
  IFOVPF  421C    DECK-ID 41F  FORTRAN 3.3B          SUMMARY-102
  PACK    4233    DECK-ID 09F  FORTRAN 3.3B          SUMMARY-102
  IOPRBD  4258    DECK-ID 31F  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  47F0    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3F1,,,B
IN

*K,I6
IN

*P
  FTN33B  2DD3    DECK-ID 01F  FORTRAN 3.3B          SUMMARY-102
  ERRMSG  3A85    DECK-ID 42F  FORTRAN 3.3B          SUMMARY-102
  IOPRBD  4A0F    DECK-ID 31F  FORTRAN 3.3B          SUMMARY-102
  ENDLOC  4FA7    DECK-ID 17F  FORTRAN 3.3B          SUMMARY-102
IN

*K,I8
IN

*N,FTN3ER,,,B
IN

*K,I6
IN

*L,READ
IN

*L,QBPREP
IN

*L,QBQF2I
IN

*L,ABS
IN
```

```
*L,SQRT
IN

*L,SIGN
IN

*L,FLOAT
IN

*L,EXP
IN

*L,ALOG
IN

*L,TANH
IN

*L,SIN
IN

*L,ATAN
IN

*L,PARABS
IN

*L,Q8IFRM
IN

*L,Q8FS
IN

*L,Q8TRAN
IN

*L,Q8QINI
IN

*L,Q8QEND
IN

*L,Q8CMPD
IN

*L,Q8RWBU
IN

*L,Q8ERRM
IN

*L,Q8DFNF
IN

*L,Q8QX
IN

*L,Q8QUN1
IN

*L,Q8FGET
IN

*L,Q8MAGT
IN

*L,EUF
IN

*L,IOCK
IN

*L,Q8PSE
IN

*L,Q8PAND
IN

*L,Q8EXP1
IN

*L,Q8EXP9
IN

*L,SETBFR
IN

*L,ENCODE
IN
```

```
*L,COMMON
IN

*L,IGETCH
IN

*L,IPACK
IN

*L,UPDATE
IN

*L,DECPL
IN

*L,INTGR
IN

*L,SPACEX
IN

*L,HOLRTH
IN

*L,DCHX
IN

*L,HXASC
IN

*L,AFRMOT
IN

*L,RFRMOT
IN

*L,AFRMIN
IN

*L,RFRMIN
IN

*L,ASCHX
IN

*L,HXDC
IN

*L,FLOTIN
IN

*L,FOUT
IN

*L,EOUT
IN

*L,EWRITE
IN

*L,INITL1
IN

*L,FORMTR
IN

*L,Q80FI
IN

*L,Q80FL
IN

*L,Q80FX
IN

*L,HEXASC
IN

*L,HEXDEC
IN

*L,ASCII
IN

*L,DECHEX
IN

*L,AFORM
IN
```

```
*L,RFORM
IN

*L,FLOATG
IN

*L,FLOT
IN

*L,OPERND
IN

*L,08QD2I
IN

*L,SNGL
IN

*L,DABS
IN

*L,DSQRT
IN

*L,DSIGN
IN

*L,DEXP
IN

*L,DLOG
IN

*L,DSIN
IN

*L,DATAN
IN

*L,08DXP1
IN

*L,08DXP9
IN

*L,08QDFI
IN

*L,DOUT
IN

*L,DFLOT
IN

*L,DRSTOR
IN

*V
IN

*V        RPGII
IN

*V
IN

*K,I6
IN

*L,RPGII
IN

*K,P8
IN

*P,,,R9BASE
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SAY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIA    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVA    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN
```

```
*N,RPGSM0,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIH   3A61    DECK-ID C01  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM1,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIF   3A61    DECK-ID C02  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM2,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIE   3A61    DECK-ID C03  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM3,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIL   3A61    DECK-ID C04  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN
```

```
*N,RPGSM4,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIII   3A61    DECK-ID C05  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM5,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIC   3A61    DECK-ID C06  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM6,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIO   3A61    DECK-ID C07  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RPGSM7,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0          SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0          SUMMARY-108
  R9LBY    2DF2    DECK-ID R55  RPGII 1.0          SUMMARY-108
  R9SBY    2DFF    DECK-ID R97  RPGII 1.0          SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0          SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0          SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0          SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0          SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0          SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0          SUMMARY-108
  RPGIIA   3A61    DECK-ID C08  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN
```

```
*N,RPGSM8,,,B
IN

*K,I6
IN

*P,,,OVLYPT
  RPG      2DD3    DECK-ID R01  RPGII 1.0            SUMMARY-108
  RPGDMY   2DF0    DECK-ID C11  RPGII 1.0            SUMMARY-108
  R9LRY    2DF2    DECK-ID R55  RPGII 1.0            SUMMARY-108
  RGSRY    2DFF    DECK-ID R97  RPGII 1.0            SUMMARY-108
  R9MIW    2E12    DECK-ID R63  RPGII 1.0            SUMMARY-108
  R9MVW    2E19    DECK-ID R73  RPGII 1.0            SUMMARY-108
  R9MIB    2E26    DECK-ID R62  RPGII 1.0            SUMMARY-108
  R9MVB    2E3A    DECK-ID R71  RPGII 1.0            SUMMARY-108
  R9ARG    2E55    DECK-ID R04  RPGII 1.0            SUMMARY-108
  RPGROT   2E6D    DECK-ID C10  RPGII 1.0            SUMMARY-108
  RPGIIM   3A61    DECK-ID C09  RPGII 1.0            SUMMARY-108
IN

*K,I8
IN

*N,RPGSM9,,,B
IN

*K,I6
IN

*L,R9ADSB
IN

*L,R9ARG
IN

*L,R9ASQR
IN

*L,R9ATON
IN

*L,R9BDPR
IN

*L,R9BINP
IN

*L,R9BINT
IN

*L,R9BITF
IN

*L,R9BITN
IN

*L,R9CALC
IN

*L,R9CHAN
IN

*L,R9CHIN
IN

*L,R9CLOS
IN

*L,R9CLRC
IN

*L,R9CLRE
IN

*L,R9CMBD
IN

*L,R9CMOV
IN

*L,R9CNTR
IN

*L,R9COMP
IN

*L,R9CRIN
IN

*L,R9DEBG
IN
```

```
*L,R9DETP
IN

*L,R9DIVD
IN

*L,R9DMND
IN

*L,R9DSPY
IN

*L,R9ECOD
IN

*L,R9EDCN
IN

*L,R9EXCP
IN

*L,R9EXIT
IN

*L,R9FILR
IN

*L,R9FINS
IN

*L,R9FLDL
IN

*L,R9FLOW
IN

*L,R9FNBG
IN

*L,R9FORC
IN

*L,R9FSTL
IN

*L,R9FTOV
IN

*L,R9GETS
IN

*L,R9GOTO
IN

*L,R9ICTL
IN

*L,R9IDMX
IN

*L,R9INDM
IN

*L,R9INIG
IN

*L,R9INIT
IN

*L,R9INMV
IN

*L,R9INTA
IN

*L,R9INTL
IN

*L,R9IOCL
IN

*L,R9IPBG
IN

*L,R9IPUT
IN

*L,R9ITLP
IN
```

```
*L,R9LAHD
IN

*L,R9LBY
IN

*L,R9LCAE
IN

*L,R9LEL
IN

*L,R9LKUP
IN

*L,R9LOAD
IN

*L,R9LOCL
IN

*L,R9LRCK
IN

*L,R9MIB
IN

*L,R9MI*
IN

*L,R9MMOV
IN

*L,R9MOVA
IN

*L,R9MOVE
IN

*L,R9MOVZ
IN

*L,R9MTRN
IN

*L,R9MT*K
IN

*L,R9MULT
IN

*L,R9MVB
IN

*L,R9MVTA
IN

*L,R9MVW
IN

*L,R9NRMX
IN

*L,R9NSQR
IN

*L,R9NTOA
IN

*L,R9NXFL
IN

*L,R9NXRC
IN

*L,R9OPNF
IN

*L,R9OTMG
IN

*L,R9OTMV
IN

*L,R9OTPT
IN

*L,R9OV50
IN
```

```
*L,R9OVOP
IN

*L,R9PACK
IN

*L,R9PAGE
IN

*L,R9POSS
IN

*L,R9PRCL
IN

*L,R9PRFN
IN

*L,R9PUTS
IN

*L,R9RCAD
IN

*L,R9RDEN
IN

*L,R9READ
IN

*L,R9RPGO
IN

*L,R9RPRT
IN

*L,R9RSLT
IN

*L,R9RSTS
IN

*L,R9SBY
IN

*L,R9SETF
IN

*L,R9SETN
IN

*L,R9SHFT
IN

*L,R9SKIP
IN

*L,R9SPAC
IN

*L,R9SPTP
IN

*L,R9SQRT
IN

*L,R9STH0
IN

*L,R9STLL
IN

*L,R9STON
IN

*L,R9TBOT
IN

*L,R9TIME
IN

*L,R9TP40
IN

*L,R9TRAL
IN

*L,R9TRCE
IN
```

```
*L,R9TROT
IN

*L,R9TSTB
IN

*L,R9TSTN
IN

*L,R9TSTZ
IN

*L,R9TTOP
IN

*L,R9UNPK
IN

*L,R9XCPT
IN

*L,R9XFOT
IN

*L,R9XRSD
IN

*L,R9YCOD
IN

*L,R9ZADS
IN

*L,R9ZCOD
IN

*L,DFUT
IN

*P
  DFUT     3202    DECK-ID U12  RPGII 1.0          SUMMARY-108
  DUE      3219    DECK-ID U13  RPGII 1.0          SUMMARY-108
  DECHEX   35A7    DECK-ID U11  RPGII 1.0          SUMMARY-108
  BINASC   35FA    DECK-ID U03  RPGII 1.0          SUMMARY-108
  ATOM     3659    DECK-ID U02  RPGII 1.0          SUMMARY-108
  FATMGR   3603    DECK-ID U14  RPGII 1.0          SUMMARY-108
  MOVMYT   3859    DECK-ID U46  RPGII 1.0          SUMMARY-108
  CMSG     3MCB    DECK-ID U37  RPGII 1.0          SUMMARY-108
  HELP     3908    DECK-ID U15  RPGII 1.0          SUMMARY-108
  MM       3A96    DECK-ID U17  RPGII 1.0          SUMMARY-108
  MOTION   3AC7    DECK-ID U18  RPGII 1.0          SUMMARY-108
  BINHEX   3AF9    DECK-ID U04  RPGII 1.0          SUMMARY-108
  ERRMSG   3B36    DECK-ID U42  RPGII 1.0          SUMMARY-108
  DEFINE   3B97    DECK-ID U39  RPGII 1.0          SUMMARY-108
  AUDIT    3D91    DECK-ID U32  RPGII 1.0          SUMMARY-108
  DUMP     3FD7    DECK-ID U41  RPGII 1.0          SUMMARY-108
  INIT     4117    DECK-ID U44  RPGII 1.0          SUMMARY-108
  LOAD     4313    DECK-ID U45  RPGII 1.0          SUMMARY-108
  COPY     4519    DECK-ID U38  RPGII 1.0          SUMMARY-108
  DISCRD   47A6    DECK-ID U40  RPGII 1.0          SUMMARY-108
  ADROUT   4873    DECK-ID U30  RPGII 1.0          SUMMARY-108
  PURGE    49CH    DECK-ID U47  RPGII 1.0          SUMMARY-108
  SAVE     4AE9    DECK-ID U51  RPGII 1.0          SUMMARY-108
  RELOAD   4C38    DECK-ID U50  RPGII 1.0          SUMMARY-108
  TAPMGR   4E41    DECK-ID U22  RPGII 1.0          SUMMARY-108
  ADRPRG   501A    DECK-ID U31  RPGII 1.0          SUMMARY-108
  ADRSKL   510A    DECK-ID U01  RPGII 1.0          SUMMARY-108
  FORTN    523A    DECK-ID U54  RPGII 1.0          SUMMARY-108
  QBPRMS   536F    DECK-ID U55  RPGII 1.0          SUMMARY-108
IN

*K,I6
IN

*N,DISKUT,,,B
IN

*K,I6
IN

*L,RPGFIL
IN

*L,CATLOG
IN

*K,PB
IN
```

```
*P,,,CATSEG
  CATLOG  2DD3    DECK-ID U06  RPGII 1.0          SUMMARY-108
  CATFIL  2DE4    DECK-ID U34  RPGII 1.0          SUMMARY-108
  CATGET  30A5    DECK-ID U05  RPGII 1.0          SUMMARY-108
  CATSKL  30C5    DECK-ID U07  RPGII 1.0          SUMMARY-108
  CATOBJ  32C6    DECK-ID U36  RPGII 1.0          SUMMARY-108
  CATERR  3340    DECK-ID U33  RPGII 1.0          SUMMARY-108
  CATMSG  33F8    DECK-ID U35  RPGII 1.0          SUMMARY-108
  FORTN   3472    DECK-ID U54  RPGII 1.0          SUMMARY-108
  QBPRMS  35A7    DECK-ID U55  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,CATFIL,,,B
IN

*K,I6
IN

*L,SWITCH
IN

*K,P8
IN

*P,,,SWISEG
  SWITCH  2DD3    DECK-ID U21  RPGII 1.0          SUMMARY-108
  SWIFIL  2DE4    DECK-ID U53  RPGII 1.0          SUMMARY-108
  FORTN   2F1F    DECK-ID U54  RPGII 1.0          SUMMARY-108
  QBPRMS  3054    DECK-ID U55  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,SWIFIL,,,B
IN

*K,I6
IN

*L,RRDPCH
IN

*K,P8
IN

*P,,,RRDSEG
  RRDPCH  2DD3    DECK-ID U24  RPGII 1.0          SUMMARY-108
  RRDFIL  2DE4    DECK-ID U46  RPGII 1.0          SUMMARY-108
  RRPERR  311E    DECK-ID U49  RPGII 1.0          SUMMARY-108
  GETLPG  31EE    DECK-ID U43  RPGII 1.0          SUMMARY-108
  SFCGET  3268    DECK-ID U52  RPGII 1.0          SUMMARY-108
  FORTN   3288    DECK-ID U54  RPGII 1.0          SUMMARY-108
  QBPRMS  33BD    DECK-ID U55  RPGII 1.0          SUMMARY-108
IN

*K,I8
IN

*N,RRDFIL,,,B
IN

*K,I6
IN

*L,OPEN01
IN

*L,OPEN02
IN

*L,OPEN03
IN

*L,OPEN04
IN

*L,OPEN05
IN

*L,OPEN06
IN

*L,OPEN07
IN

*L,READ09
IN
```

```
*L*READ10
IN

*L*READ11
IN

*L*READ12
IN

*L*READ13
IN

*L*READ14
IN

*L*READ15
IN

*L*WRIT17
IN

*L*WRIT18
IN

*L*WRIT19
IN

*L*WRIT20
IN

*L*WRIT21
IN

*L*WRIT22
IN

*L*WRIT23
IN

*L*UPDT25
IN

*L*UPDT26
IN

*L*UPDT27
IN

*L*UPDT28
IN

*L*UPDT29
IN

*L*ADDT30
IN

*L*ADDT31
IN

*L*ADDT32
IN

*L*SETL33
IN

*L*SLCT34
IN

*L*CLOS35
IN

*L*CLOS36
IN

*L*CLOS37
IN

*L*CLOS38
IN

*L*CLOS39
IN

*L*CLOS40
IN

*L*CLOS41
IN
```

```
*L,ROOT43
IN

*L,ROOT44
IN

*L,ROOT45
IN

*L,ERR046
IN

*L,NTAP48
IN

*L,NTAP49
IN

*L,SORT50
IN

*L,CKEY51
IN

*L,NTAP52
IN

*L,NTAP53
IN

*L,NTAP54
IN

*L,NTAP55
IN

*L,NTAP57
IN

*L,NTAP58
IN

*L,MOUNT
IN

*L,I000WR
IN

*L,CM00PT
IN

*L,CM02IN
IN

*L,CM03GO
IN

*L,STRACE
IN

*V  DEBUGGING AND CHECKOUT
IN

*K,I6
IN

*L,TRACE
IN

*K,I6
IN

*K,P8
IN

*P,F
  SETBR1  2DD3    DECK-ID M91  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPST,,,8
IN

*K,I6
IN

*K,P8
IN
```

```
*P,F
  TERMI1   2DD3     DECK-ID M92  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPCLR,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  ENTCO1   2DD3     DECK-ID M93  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,PPLOAD,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  PESUM1   2DD3     DECK-ID M94  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPEND,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  PWTRE1   2DD3     DECK-ID M95  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPRLST,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  SETAQ1   2DD3     DECK-ID M96  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPRSET,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  CONDM1   2DD3     DECK-ID M97  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPDMPC,,,B
IN
```

```
*K,I6
IN

*K,P8
IN

*P,F
  JUMPR1  2DD3    DECK-ID M98  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,EPJMP,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  LUCHG1  2DD3    DECK-ID M99  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPBPLU,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  BPTAP1  2DD3    DECK-ID N02  MSOS 5.0              SUMMARY-110
IN

*K,I6
IN

*N,BPTAPC,,,B
IN

*K,I6
IN

*K,P8
IN

*P,F
  MASSM1  2DD3    DECK-ID N03  MSOS 5.0              SUMMARY-110
IN

*K,I8
IN

*N,BPMASS,,,B
IN

*V  SYSTEM UTILITY PROGRAMS
IN

*K,I6
IN

*L,LULIST
IN

*L,LISTH
IN

*L,OPSORT
IN

*L,EESORT
IN

*L,COSY
IN

*L,LCOSY
IN

*L,CYFT
IN
```

```
*L,IOUP
IN

*K,P6
IN

*P,F
  IOUP    2DD3    DECK-ID N82  MSOS 5.0              SUMMARY-110
  IOUPV4  2DE5    DECK-ID N83  MSOS 5.0              SUMMARY-110
  NXTLOC  3890    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,IOUPV4,,,8
IN

*K,I6
IN

*K,P8
IN

*L,MTUP
IN

*P,F,,TAPUTL
  MTUP    2DD3    DECK-ID U01  MAG TAPE UTILITY      SUMMARY-106
  TAPUTL  2DEE    DECK-ID U02  MAG TAPE UTILITY      SUMMARY-106
  FNN     2FE4    DECK-ID U03  MAG TAPE UTILITY      SUMMARY-106
  SCAN    3091    DECK-ID U04  MAG TAPE UTILITY      SUMMARY-106
  OPENIO  3191    DECK-ID U05  MAG TAPE UTILITY      SUMMARY-106
  RDWTH   3387    DECK-ID U06  MAG TAPE UTILITY      SUMMARY-106
  LIOC    344A    DECK-ID U07  MAG TAPE UTILITY      SUMMARY-106
  COPY    3540    DECK-ID U08  MAG TAPE UTILITY      SUMMARY-106
  EXIT    357C    DECK-ID U09  MAG TAPE UTILITY      SUMMARY-106
  PRINT   35A3    DECK-ID U10  MAG TAPE UTILITY      SUMMARY-106
  VERIFY  373F    DECK-ID U11  MAG TAPE UTILITY      SUMMARY-106
  SELECT  3859    DECK-ID U12  MAG TAPE UTILITY      SUMMARY-106
  PRINTI  39BC    DECK-ID U13  MAG TAPE UTILITY      SUMMARY-106
  DUMP    3A32    DECK-ID U14  MAG TAPE UTILITY      SUMMARY-106
  INIT    3C11    DECK-ID U15  MAG TAPE UTILITY      SUMMARY-106
  STNLAB  3CAD    DECK-ID U16  MAG TAPE UTILITY      SUMMARY-106
  CVASEB  403D    DECK-ID U17  MAG TAPE UTILITY      SUMMARY 110
  ALCBUF  414F    DECK-ID U18  MAG TAPE UTILITY      SUMMARY-106
  NXTLOC  41BD    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,MTUPFL,,,8
IN

*K,I6
IN

*L,DTLP
IN

*K,P8
IN

*P,F
  DSKTAP  2DD3    DECK-ID N90  MSOS 5.0              SUMMARY-110
  DSKEQC  3058    DECK-ID N91  MSOS 5.0              SUMMARY-110
  DSKDMX  30D9    DECK-ID 068  MSOS 5.0              SUMMARY-110
  DSKCDR  317E    DECK-ID D11  PERIPH. DRIVERS 1.1C  SUMMARY-110
  DTLCT9  3224    DECK-ID C46  PERIPH. DRIVERS 1.1C  SUMMARY-110
  MDRSMD  3582    DECK-ID C94  PERIPH. DRIVERS 1.1C  SUMMARY-110
  NXTLOC  390D    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,DSKTAP,,,8
IN

*K,I6
IN

*L,SETPV4
IN

*K,P8
IN
```

```
*P,F
  SPOLY1  2DD3    DECK-ID N93  MSOS 5.0              SUMMARY-110
  STPV4   2DED    DECK-ID N94  MSOS 5.0              SUMMARY-110
  IERROR  2E49    DECK-ID N95  MSOS 5.0              SUMMARY-110
  MCTOK   2E9C    DECK-ID N96  MSOS 5.0              SUMMARY-110
  GETPAG  2EAC    DECK-ID N97  MSOS 5.0              SUMMARY-110
  CONPRT  2EBB    DECK-ID N98  MSOS 5.0              SUMMARY-110
  REDCON  2EFE    DECK-ID N99  MSOS 5.0              SUMMARY-110
  CONDEC  2F56    DECK-ID 001  MSOS 5.0              SUMMARY-110
  OKDERM  30F7    DECK-ID 002  MSOS 5.0              SUMMARY-110
  IREAD   3392    DECK-ID 003  MSOS 5.0              SUMMARY-110
  ASCOUT  3428    DECK-ID 004  MSOS 5.0              SUMMARY-110
  PARAMS  3455    DECK-ID 005  MSOS 5.0              SUMMARY-110
  DISKIO  3461    DECK-ID 006  MSOS 5.0              SUMMARY-110
  NXTLOC  348C    NEXT AVAILABLE LOCATION
IN

*K,I6
IN

*N,STP1V4,,,8
IN

*K,I6
IN

*K,P8
IN

*P,F
  SPOLY2  2DD3    DECK-ID 007  MSOS 5.0              SUMMARY-110
  SUP     2DE5    DECK-ID 008  MSOS 5.0              SUMMARY-110
  IERROR  30C9    DECK-ID N95  MSOS 5.0              SUMMARY-110
  GETPAG  311C    DECK-ID N97  MSOS 5.0              SUMMARY-110
  ATOA    3128    DECK-ID 009  MSOS 5.0              SUMMARY-110
  ISTAT   3182    DECK-ID 010  MSOS 5.0              SUMMARY-110
  SCIO    3186    DECK-ID 011  MSOS 5.0              SUMMARY-110
  SCRD    31C5    DECK-ID 012  MSOS 5.0              SUMMARY-110
  REDCON  31D4    DECK-ID N99  MSOS 5.0              SUMMARY-110
  ICAT    322C    DECK-ID 013  MSOS 5.0              SUMMARY-110
  BUFIN   3348    DECK-ID 014  MSOS 5.0              SUMMARY-110
  MOVE    3422    DECK-ID 015  MSOS 5.0              SUMMARY-110
  IREAD   35BB    DECK-ID 003  MSOS 5.0              SUMMARY-110
  ASCOUT  3621    DECK-ID 004  MSOS 5.0              SUMMARY-110
  PARAMS  364E    DECK-ID 005  MSOS 5.0              SUMMARY-110
  DISKIO  365A    DECK-ID 006  MSOS 5.0              SUMMARY-110
  NXTLOC  3685    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,STP2V4,,,8
IN

*V  INSTALL LIBRARY BUILDER
IN

*K,I6
IN

*L,LIBILD
IN

*K,P8
IN

*P,F
  LIBIDO  2DD3    DECK-ID 030  MSOS 5.0              SUMMARY-110
  CONVRS  2DF0    DECK-ID 036  MSOS 5.0              SUMMARY-110
  MESSY   3065    DECK-ID 031  MSOS 5.0              SUMMARY-110
  LJA2B   30DE    DECK-ID 037  MSOS 5.0              SUMMARY-110
  MOVECH  3128    DECK-ID 032  MSOS 5.0              SUMMARY-110
  PICKUP  317C    DECK-ID 033  MSOS 5.0              SUMMARY-110
  IOSUB   3195    DECK-ID 034  MSOS 5.0              SUMMARY-110
  NXTLOC  3189    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,LIBIDO,,,8
IN

*K,I6
IN

*K,P8
IN
```

```
*P,F
  HELPER  2DD3    DECK-ID 035  MSOS 5.0              SUMMARY-110
  MOVECH  306D    DECK-ID 032  MSOS 5.0              SUMMARY-110
  HELP0   310E    DECK-ID 039  MSOS 5.0              SUMMARY-110
  HELP1   3726    DECK-ID 040  MSOS 5.0              SUMMARY-110
  HELP2   3779    DECK-ID 041  MSOS 5.0              SUMMARY-110
  HELP3   37C0    DECK-ID 042  MSOS 5.0              SUMMARY-110
  HELP4   381E    DECK-ID 043  MSOS 5.0              SUMMARY-110
  HELP5   3A85    DECK-ID 044  MSOS 5.0              SUMMARY-110
  HELP8   3C4E    DECK-ID 045  MSOS 5.0              SUMMARY-110
  HELP9   3D58    DECK-ID 046  MSOS 5.0              SUMMARY-110
  HELP10  3E17    DECK-ID 047  MSOS 5.0              SUMMARY-110
  HELP11  3E8A    DECK-ID 048  MSOS 5.0              SUMMARY-110
  HELP12  3EDA    DECK-ID 049  MSOS 5.0              SUMMARY-110
  HELP13  40D7    DECK-ID 050  MSOS 5.0              SUMMARY-110
  HELP14  4139    DECK-ID 036  MSOS 5.0              SUMMARY-110
  NXTLOC  4144    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,HELPER,,,8
IN

*V  INSTALL SKELETON EDITOR
IN

*K,I6
IN

*L,SKED
IN

*K,P8
IN

*P,F
  SKFILE  2DD3    DECK-ID 052  MSOS 5.0              SUMMARY-110
  NXTLOC  3E5E    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SKFILE,,,8
IN

*V  SYSTEM INITIALIZER
IN

*K,I6
IN

*L,SILP
IN

*K,P8
IN

*P,F
  CONTRL  2DD3    DECK-ID 017  MSOS 5.0              SUMMARY-110
  ILOAD   3704    DECK-ID 016  MSOS 5.0              SUMMARY-110
  LDRTBL  3063    DECK-ID 019  MSOS 5.0              SUMMARY-110
  I1      3F7F    DECK-ID 020  MSOS 5.0              SUMMARY-110
  I2      4185    DECK-ID 021  MSOS 5.0              SUMMARY-110
  Q1810   41A5    DECK-ID C26  PERIPH. DRIVERS 1.0C  SUMMARY-106
  Q1627   426F    DECK-ID 006  PERIPH. DRIVERS 1.1C  SUMMARY-110
  IDRIV   4209    DECK-ID 023  MSOS 5.0              SUMMARY-110
  GMLCT9  434D    DECK-ID C44  PERIPH. DRIVERS 1.1C  SUMMARY-110
  MDRIV   4699    DECK-ID 022  MSOS 5.0              SUMMARY-110
  Q16331  46A1    DECK-ID C86  PERIPH. DRIVERS 1.1C  SUMMARY-110
  GCDDMY  4C77    DECK-ID C49  PERIPH. DRIVERS 1.1C  SUMMARY-110
  GPTDMY  4C79    DECK-ID 005  PERIPH. DRIVERS 1.1C  SUMMARY-110
  NXTLOC  4C7B    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,SI,,,8
IN

*K,I6
IN

*L,SMDMPI
IN
```

```
*K,P8
IN

*P
  SMDINT  2DD3    DECK-ID C93  PERIPH. DRIVERS 1.1C  SUMMARY-110
  NXTLOC  2EA8    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,MPSMDI,,,8
IN

*K,I6
IN

*L,SMDMPT
IN

*K,P8
IN

*P
  SMDTGS  2DD3    DECK-ID C95  PERIPH. DRIVERS 1.1C  SUMMARY-110
  NXTLOC  473F    NEXT AVAILABLE LOCATION
IN

*K,I8
IN

*N,MPSMDT,,,8
IN

*Z
*K,I10,P11,L9
*CTO,     MSOS 5.0 INSTALLATION COMPLETED - YOU MAY AUTOLOAD
*Z
```

```
*LIBEDT
 LIB

 IN

 *DM
  1  0030  23F5  0000  28A4  04R5  0000  02H0
  2  0000  0000  0000  0000  0CC9  0000  02EF
  3  0010  281A  0000  C45D  0285  0000  0312
  4  0020  28E5  0000  000E  0233  0000  031A
  5  0030  23F5  0000  26FA  04EE  0000  0320
  6  0020  28E5  0000  0000  01A3  0000  0330
  7  0020  28E5  0000  0000  0193  0000  0335
  8  0020  28E5  0000  3202  018F  0000  033A
  9  0020  28E5  0000  0000  0162  0000  033F
 10  0020  28E5  0000  0003  00F2  0000  0343
 11  0020  28E5  0000  0000  0062  0000  0346
 12  0030  23F5  0000  000E  02E2  0000  0348
 13  0030  0000  0000  0000  026F  0000  0350
 14  0030  0000  0000  0000  036E  0000  0357
 15  0020  0000  0000  0000  00D7  0000  0361
 16  0030  0000  0000  0000  034D  0000  0364
 17  0030  0000  0000  0000  049B  0000  036D
 18  0040  0000  0000  0000  4500  0000  037A
 19  0040  0000  0000  0000  0192  0000  0432
 20  0040  0000  0000  0000  1560  0000  0437
 21  0043  18F3  0000  C45D  0258  0000  0470
 22  0044  18F3  0000  0001  0160  0000  0477
 23  0044  18F3  0000  CE02  019D  0000  0478
 24  0040  0000  0000  0000  0A9E  0000  0480
 25  0044  18F3  0000  1607  0D35  0000  273B
 26  0040  0000  0000  0000  0000  0000  04A2
 27  0040  0000  0000  0000  0000  0000  04A2
 28  0040  0000  0000  0000  0000  0000  04A2
 29  0040  0000  0000  0000  0000  0000  04A2
 30  0040  0000  0000  0000  0000  0000  04A2
 31  0040  0000  0000  0000  0000  0000  04A2
 32  0040  0000  0000  0000  0000  0000  04A2
 33  0040  0000  0000  0000  0000  0000  04A2
 34  0040  0000  0000  0000  0000  0000  04A2
 35  0040  0000  0000  0000  0000  0000  04A2

FINI
IN

*DL
LIBMAC  SECT.     1278
ASSEM   SECT.     1294
```

```
ASSIM   SECT.   1294
PASS1   SECT.   129A  FILE
PASS2   SECT.   12C6  FILE
PASS3   SECT.   12E0  FILE
TABLST  SECT.   1305  FILE
XREF    SECT.   131A  FILE
MACSKL  SECT.   132B  FILE
MACHOS  SECT.   159C  FILE
EDITOR  SECT.   15AD
EDITFL  SECT.   15B1  FILE
SMC     SECT.   15D8
SMCMON  SECT.   15DD  FILE
SMCEDT  SECT.   15FF  FILE
SMCSRT  SECT.   1611  FILE
SMCIMG  SECT.   1617  FILE
SMCFMG  SECT.   161D  FILE
FTN     SECT.   1624
EXITF   SECT.   1624
PAGCHK  SECT.   1624
ASCOPT  SECT.   1624
PRGNAM  SECT.   1624
PAGNBR  SECT.   1624
DATE    SECT.   1624
TIME    SECT.   1624
FTN3A1  SECT.   162E  FILE
FTN3A2  SECT.   169E  FILE
FTN3A3  SECT.   16C6  FILE
FTN3A4  SECT.   16F1  FILE
FTN3A5  SECT.   171A  FILE
FTN3B1  SECT.   173D  FILE
FTN3C1  SECT.   17BF  FILE
FTN3D1  SECT.   1831  FILE
FTN3E1  SECT.   1891  FILE
FTN3F1  SECT.   18F6  FILE
FTN3ER  SECT.   193C  FILE
READ    SECT.   1998
WRITE   SECT.   1998
FREAD   SECT.   1998
FWRITE  SECT.   1998
SCHEDL  SECT.   1998
TIMER   SECT.   1998
DISPAT  SECT.   1998
DISP    SECT.   1998
LINK    SECT.   1998
ICLOCK  SECT.   1998
INPINS  SECT.   1998
OUTINS  SECT.   1998
RELESE  SECT.   1998
ICONCT  SECT.   1998
OCONCT  SECT.   1998
UPPREP  SECT.   19A5
UPPKUP  SECT.   19A5
Q9PKUP  SECT.   19A5
QBQF2I  SECT.   19AB
QBQI2F  SECT.   19AB
QBQF2F  SECT.   19AB
RETAD   SECT.   19AB
QSAVE   SECT.   19AB
QBAB    SECT.   19B3
ABS     SECT.   19B3
SQRT    SECT.   19B8
QBSG    SECT.   19BF
SIGN    SECT.   19BF
QBQFIX  SECT.   19C4
QBFX    SECT.   19C4
QBQFLT  SECT.   19C4
QBFLOT  SECT.   19C4
IFIX    SECT.   19C4
FLOAT   SECT.   19C4
DFIX    SECT.   19C4
QBDFLT  SECT.   19C4
DFLT    SECT.   19C4
EXP     SECT.   19CB
ALOG    SECT.   19D3
TANH    SECT.   19DA
SIN     SECT.   19E1
COS     SECT.   19E1
ATAN    SECT.   19EA
PAPABS  SECT.   19F2
QBIFRM  SECT.   19F6
QBFS    SECT.   19FC
QBTRAN  SECT.   1A35
QBQINI  SECT.   1A81
QBUNIT  SECT.   1A81
QBSKIP  SECT.   1A81
QBQEND  SECT.   1A8B
QBCMP0  SECT.   1A90
QBCMP1  SECT.   1A90
QBDFAD  SECT.   1A90
QBQENS  SECT.   1A90
RECEND  SECT.   1A90
```

Q8RINB	SECT.	1A9B
QALOCB	SECT.	1A9B
Q6RWBU	SECT.	1A9B
Q8INTB	SECT.	1A9B
QABEGB	SECT.	1A9B
WACLRB	SECT.	1A9B
QAFINT	SECT.	1A9B
QA1MUF	SECT.	1A9B
WWFLG	SECT.	1A9B
Q8EMM4	SECT.	1AA5
QAFERM	SECT.	1AA5
Q8EREM	SECT.	1AA5
QAUFNF	SECT.	1AB1
QADFIN	SECT.	1AB1
QAQTOM	SECT.	1ABA
QAQTRM	SECT.	1ABA
WAQX	SECT.	1ABA
QAMOVE	SECT.	1ABA
QaQY	SECT.	1ABA
Q8QZ	SECT.	1ABA
QAQUN1	SECT.	1AC2
QAQUN2	SECT.	1AC2
QAQUN3	SECT.	1AC2
QAFGET	SECT.	1AC8
QAFPUT	SECT.	1AC8
QALOCF	SECT.	1AC8
Q8IGP	SECT.	1AC8
QAMAGT	SECT.	1ACE
Q8EOTT	SECT.	1ACE
QACHCK	SECT.	1AD4
QAQFLE	SECT.	1AD4
Q8QWND	SECT.	1AD4
EOF	SECT.	1AD4
IOCK	SECT.	1ADC
Q8PSE	SECT.	1AE2
Q8PSEN	SECT.	1AE2
QASTP	SECT.	1AE2
QASTPN	SECT.	1AE2
QACOMI	SECT.	1AE2
QAPAND	SECT.	1AE8
QAEXP1	SECT.	1AEF
QAEXP9	SECT.	1AF6
QAEXPT	SECT.	1AF6
QAEXP2	SECT.	1AF6
Q8GGET	SECT.	1AFF
SETBFR	SECT.	1AFF
ENCODE	SECT.	1B03
DECODE	SECT.	1B03
COMMON	SECT.	1B0A
ISAVE	SECT.	1B0A
IGETCH	SECT.	1B0E
GETCH	SECT.	1B0E
IPACK	SECT.	1B14
UPDATE	SECT.	1B1A
DECPL	SECT.	1B1E
INTGR	SECT.	1B23
SPACEX	SECT.	1B28
HOLRTH	SECT.	1B2D
QUOTE	SECT.	1B2D
DCHX	SECT.	1B35
HXASC	SECT.	1B3C
AFRMOT	SECT.	1B42
RFRMOT	SECT.	1B47
AFRMIN	SECT.	1B4C
RFRMIN	SECT.	1B52
ASCHX	SECT.	1B57
HXDC	SECT.	1B5D
FLOTIN	SECT.	1B65
FOUT	SECT.	1B6B
EOUT	SECT.	1B73
EWRITE	SECT.	1B7E
IMITL1	SECT.	1B83
RESTRE	SECT.	1B83
FORMTR	SECT.	1B89
CHCNT	SECT.	1B89
QAQFI	SECT.	1B9A
QAQFL	SECT.	1B9F
QAQFX	SECT.	1BA3
HEXASC	SECT.	1BAB
HEXDEC	SECT.	1BAD
ASCII	SECT.	1BB2
DECHEX	SECT.	1BB7
AFORM	SECT.	1BBC
RFORM	SECT.	1BC1
FLOATG	SECT.	1BC6
FLOT	SECT.	1BCB
HFLOT	SECT.	1BCB
IFALT	SECT.	1BDD
SFALT	SECT.	1BDD
OPERND	SECT.	1BDD

```
NXTOP   SECT.    1BDD
FPEROR  SECT.    1BDD
PROCHK  SECT.    1BDD
SPECOP  SECT.    1BDD
FLOFOP  SECT.    1BDD
FIXFOP  SECT.    1BDD
Q8GD2I  SECT.    1BEA
Q8OD2F  SECT.    1BEA
Q8UD2D  SECT.    1BEA
SNGL    SECT.    1BF1
DBLE    SECT.    1BF1
Q8SNGL  SECT.    1BF1
Q8DBLE  SECT.    1BF1
DABS    SECT.    1BF6
Q8DAB   SECT.    1BF6
DSQRT   SECT.    1BFB
Q8DSG   SECT.    1C04
DSIGN   SECT.    1C04
DEXP    SECT.    1C09
DLOG    SECT.    1C12
DSIN    SECT.    1C1A
DCOS    SECT.    1C1A
DATAN   SECT.    1C25
Q8DXP1  SECT.    1C2E
Q8DXP9  SECT.    1C36
Q8DXPT  SECT.    1C36
Q8DXP2  SECT.    1C36
Q8ODFI  SECT.    1C41
DOUT    SECT.    1C46
DFLOT   SECT.    1C51
HDFLOT  SECT.    1C51
DSTOR1  SECT.    1C70
RSTORI  SECT.    1C70
DSTOR2  SECT.    1C70
RPGII   SECT.    1C76
RPGSM0  SECT.    1C7A  FILE
RPGSM1  SECT.    1C9C  FILE
RPGSM2  SECT.    1CA4  FILE
RPGSM3  SECT.    1CD3  FILE
RPGSM4  SECT.    1CDA  FILE
RPGSM5  SECT.    1CDD  FILE
RPGSM6  SECT.    1CF3  FILE
RPGSM7  SECT.    1D0E  FILE
RPGSM8  SECT.    1D1E  FILE
RPGSM9  SECT.    1D2E  FILE
YSIGNI  SECT.    1D36
R9ADSB  SECT.    1D36
R9ABG   SECT.    1D3F
R9ASQR  SECT.    1D43
R9ATON  SECT.    1D48
R9BDPR  SECT.    1D4E
R9BINP  SECT.    1D57
R9BINT  SECT.    1D63
R9BITF  SECT.    1D6B
R9BITN  SECT.    1D70
R9CALC  SECT.    1D75
R9CHAN  SECT.    1D7A
R9CHIN  SECT.    1D81
R9CLOS  SECT.    1D87
R9CLRC  SECT.    1D8D
R9CLRE  SECT.    1D92
R9CMBD  SECT.    1D97
R9CMOV  SECT.    1D9C
R9CNTR  SECT.    1DA5
R9CKSG  SECT.    1DA5
R9SGTB  SECT.    1DA5
RPGSEG  SECT.    1DA5
R9COMP  SECT.    1DAA
R9CRIN  SECT.    1DB4
R9DEBG  SECT.    1DB9
R9DETP  SECT.    1DCB
YBGCYC  SECT.    1DCB
R9DIVD  SECT.    1DD1
R9BVRM  SECT.    1DD1
R9DMND  SECT.    1DDC
R9DSPY  SECT.    1DE2
R9ECOD  SECT.    1DEF
R9EDCN  SECT.    1DFB
R9EXCP  SECT.    1E04
R9EXIT  SECT.    1E09
R9FILR  SECT.    1E0E
R9FINS  SECT.    1E15
R9FLDL  SECT.    1E1A
R9FLOW  SECT.    1E20
R9FNBG  SECT.    1E25
R9FORC  SECT.    1E2B
R9FSTL  SECT.    1E2D
R9FTOV  SECT.    1E34
R9GETS  SECT.    1E3B
R9TFLG  SECT.    1E3B
R9GOTO  SECT.    1E42
```

```
R9EXSR  SECT.    1E42
R9MDSR  SECT.    1F42
R9ICTL  SECT.    1E48
R9IDMX  SECT.    1E52
R9INDM  SECT.    1F56
R9INIG  SECT.    1E5D
R9INIT  SECT.    1E60
R9INMV  SECT.    1E66
R9INTA  SECT.    1E75
R9RECP  SECT.    1E75
R9TANF  SECT.    1E75
R9INTL  SECT.    1E7D
R9IOCL  SECT.    1E83
R9IPRG  SECT.    1E87
R9IPUT  SECT.    1E8A
R9ITLP  SECT.    1F90
R9LAMD  SECT.    1E96
R9LBY   SECT.    1E9B
R9LCAE  SECT.    1E9F
R9ACAX  SECT.    1E9F
R9ACX1  SECT.    1E9F
YACAXN  SECT.    1E9F
R9LEL   SECT.    1EA7
R9GTL   SECT.    1EA7
R9LKUP  SECT.    1EAC
R9LOAD  SECT.    1FBB
R9LOCL  SECT.    1EC1
R9LRCK  SECT.    1EC6
R9MIB   SECT.    1ECC
R9MIW   SECT.    1ED1
R9MMOV  SECT.    1ED5
R9MOVA  SECT.    1FE2
R9MOVE  SECT.    1EE8
R9MOVZ  SECT.    1EF5
R9MTRN  SECT.    1EFC
R9MTWK  SECT.    1F01
R9MULT  SECT.    1F06
R9MVB   SECT.    1F0E
R9MVTA  SECT.    1F13
R9MVW   SECT.    1F1D
R9NRMX  SECT.    1F21
R9NSOR  SECT.    1F28
R9NTOA  SECT.    1F2E
R9NXFL  SECT.    1F34
R9NXRC  SECT.    1F39
R9OPNF  SECT.    1F3F
R9OTMG  SECT.    1F45
R9OTMV  SECT.    1F4A
R9OTM1  SECT.    1F4A
R9EDT1  SECT.    1F4A
R9EDT2  SECT.    1F4A
R9EDT3  SECT.    1F4A
R9EDT4  SECT.    1F4A
R9PGSW  SECT.    1F4A
R9OTPT  SECT.    1F60
R9OV50  SECT.    1F65
R9OVTB  SECT.    1F65
R9OVOP  SECT.    1F6B
R9PACK  SECT.    1F71
R9PAGE  SECT.    1F78
R9POSS  SECT.    1F7D
R9POSO  SECT.    1F7D
R9OCOU  SECT.    1F7D
R9PRCL  SECT.    1F88
R9STRT  SECT.    1F88
R9NXTA  SECT.    1F8B
R9PRFN  SECT.    1F9A
R9PUTS  SECT.    1F9F
R9CVEA  SECT.    1F9F
R9RCAD  SECT.    1FA7
R9RDEN  SECT.    1FAF
R9READ  SECT.    1FB7
R9RPGO  SECT.    1FBC
R9RPRT  SECT.    1FC1
R9INTB  SECT.    1FC1
R9CLIN  SECT.    1FC1
R9RTIN  SECT.    1FC1
R9VIND  SECT.    1FC1
R9USND  SECT.    1FC1
R9MRIN  SECT.    1FC1
R9FCTR  SECT.    1FC1
R9STTS  SECT.    1FC1
R9HLFJ  SECT.    1FC1
R9REPT  SECT.    1FC1
R9RYST  SECT.    1FC1
R9ACC1  SECT.    1FC1
YACC1B  SECT.    1FC1
YACC10  SECT.    1FC1
R9AC1S  SECT.    1FC1
R9AC2S  SECT.    1FC1
```

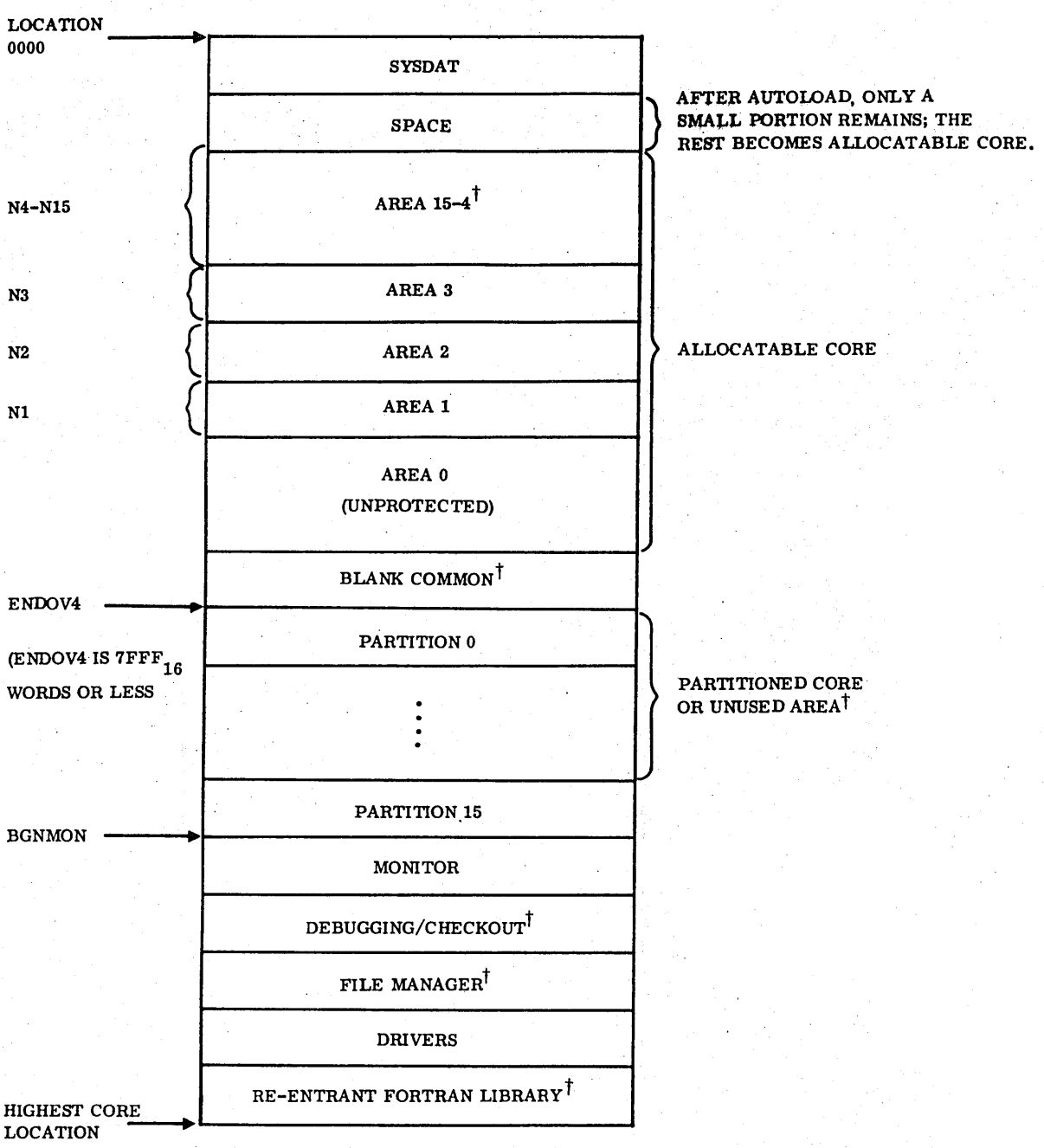
```
R9AC3S  SECT.     1FC1
R9AC1N  SECT.     1FC1
R9AC2N  SECT.     1FC1
R9AC3N  SECT.     1FC1
R9PPUP  SECT.     1FC1
R9UPOP  SECT.     1FC1
R9INRZ  SECT.     1FC1
R9INNP  SECT.     1FC1
R9INRM  SECT.     1FC1
R9PONT  SECT.     1FC1
R9CNCL  SECT.     1FC1
YERRSW  SECT.     1FC1
R9FTSW  SECT.     1FC1
R9VSWT  SECT.     1FC1
R9OVSW  SECT.     1FC1
R9FTIM  SECT.     1FC1
R9FFCB  SECT.     1FC1
R9CFIL  SECT.     1FC1
R9NFCB  SECT.     1FC1
R9PRGD  SECT.     1FC1
R9MRSW  SECT.     1FC1
R9MRPR  SECT.     1FC1
R9LRSW  SECT.     1FC1
R9BYPS  SECT.     1FC1
R9RLTR  SECT.     1FC1
R9FRMK  SECT.     1FC1
R9RSLT  SECT.     1FCF
R9RSTS  SECT.     1FD5
R9SHY   SECT.     1FDA
R9SETF  SECT.     1FDE
R9SETN  SECT.     1FE3
R9SHFT  SECT.     1FE8
YLSPAR  SECT.     1FE8
R9SKIP  SECT.     1FEF
R9SPAC  SECT.     1FF5
R9SPTP  SECT.     1FFB
R9SWRT  SECT.     2001
R9STHO  SECT.     2008
R9STLL  SECT.     2012
R9STON  SECT.     2018
R9THOT  SECT.     201D
R9TIME  SECT.     202H
R9TP40  SECT.     2030
R9TRAL  SECT.     2035
R9TRCE  SECT.     203B
R9THOT  SECT.     204D
R9TSTB  SECT.     2051
R9TSTN  SECT.     2057
R9TSTZ  SECT.     205C
R9TTOP  SECT.     2062
R9UNPK  SECT.     2067
R9XCPT  SECT.     206E
R9XFOT  SECT.     2073
R9XRSD  SECT.     207A
R9YCOD  SECT.     207F
R9ZADS  SECT.     2085
R9ZCOD  SECT.     208B
DFUT    SECT.     2091
DISKUT  SECT.     2095  FILE
RPGOBG  SECT.     20FA
R9OVER  SECT.     20FA
RPGFIL  SECT.     20FA
CATLOG  SECT.     20FE
CATSEG  SECT.     20FE
CATFIL  SECT.     2102  FILE
SWITCH  SECT.     2118
SWISEG  SECT.     2118
SWIFIL  SECT.     211C  FILE
RRDPCH  SECT.     2123
RRDSEG  SECT.     2123
RRDFIL  SECT.     2128  FILE
OPEN01  SECT.     2139
OPEN08  SECT.     2139
OPEN02  SECT.     2148
OPEN03  SECT.     2154
OPEN04  SECT.     2161
OPEN05  SECT.     218B
OPEN06  SECT.     21AF
OPEN07  SECT.     21E4
READ09  SECT.     2219
READ16  SECT.     2219
READ10  SECT.     2225
READ11  SECT.     2231
READ12  SECT.     223F
READ13  SECT.     224D
READ14  SECT.     2259
READ15  SECT.     2266
WRIT17  SECT.     2272
WRIT24  SECT.     2272
WRIT18  SECT.     227D
WRIT19  SECT.     2287
```

```
WRIT20  SECT.   2294
WRIT21  SECT.   229E
WRIT22  SECT.   22A7
WRIT23  SECT.   22B2
UPDT25  SECT.   22BC
UPDT26  SECT.   22C6
UPDT27  SECT.   22D1
UPDT28  SECT.   22DA
UPDT29  SECT.   22E4
ADDT30  SECT.   22EE
ADDT31  SECT.   22F8
ADDT32  SECT.   2303
SETL33  SECT.   230E
SLCT34  SECT.   2318
CLOS35  SECT.   231F
CLOS42  SECT.   231F
CLOS36  SECT.   2327
CLOS37  SECT.   2330
CLOS38  SECT.   2339
CLOS39  SECT.   2346
CLOS40  SECT.   2352
CLOS41  SECT.   235F
POOT43  SECT.   236C
DMPTLK  SECT.   236C
KYAREA  SECT.   236C
POOT44  SECT.   2374
POOT45  SECT.   2382
ERRO46  SECT.   238E
ERRPO   SECT.   238E
BINASC  SECT.   238E
BIZASC  SECT.   238E
BINHEX  SECT.   238E
NTAP48  SECT.   2390
NTAP49  SECT.   23B4
SORT50  SECT.   23BC
CKEY51  SECT.   23D9
NTAP52  SECT.   23DF
NTAP53  SECT.   23E7
NTAP54  SECT.   23EE
NTAP56  SECT.   23EE
NTAP55  SECT.   23F6
NTAP57  SECT.   23FC
NTAP58  SECT.   2400
MOUNT   SECT.   2407
IOOOWR  SECT.   241F
IOOOWO  SECT.   241F
CMOOPT  SECT.   2425
CMOOGT  SECT.   2425
CMO2IN  SECT.   242A
DELIM   SECT.   242A
CMO3GO  SECT.   2432
STRACE  SECT.   2437
RTRTJ   SECT.   243C
TRACE   SECT.   243C
TRACE2  SECT.   243C
TRACE1  SECT.   243C
BPST    SECT.   2475  FILE
BPCLR   SECT.   2478  FILE
BPLOAD  SECT.   247A  FILE
BPEND   SECT.   247E  FILE
BPRLST  SECT.   2481  FILE
BPPSET  SECT.   2482  FILE
BPDMPC  SECT.   2484  FILE
BPJMP   SECT.   2489  FILE
BPBPLU  SECT.   248B  FILE
BPTAPC  SECT.   248D  FILE
BPMASS  SECT.   248F  FILE
LOLIST  SECT.   2496
LISTR   SECT.   24BE
OPSORT  SECT.   24CA
BGNRD   SECT.   24CA
EESORT  SECT.   24EA
COSY    SECT.   2502
LCOSY   SECT.   2551
CYPT    SECT.   255C
IOUP    SECT.   2589
IOUPV4  SECT.   2560  FILE
MTUP    SECT.   258A
MTUPFL  SECT.   258E  FILE
DTLP    SECT.   25C3
DSKTAP  SECT.   25CA  FILE
SETPV4  SECT.   25E8
STP1V4  SECT.   25EC  FILE
STP2V4  SECT.   25FE  FILE
LIBILD  SECT.   2616
LIBIDO  SECT.   261A  FILE
HELPER  SECT.   2626  FILE
SKED    SECT.   265A
SKFILE  SECT.   265E  FILE
SILP    SECT.   268B
```

```
SI       SECT.     2692  FILE
SMDMPI   SECT.     26E4
MPSMDI   SECT.     26EC  FILE
SMDMPT   SECT.     26EF
MPSMDT   SECT.     26F7  FILE

FINI
IN

*Z
*U
MSOS 5.0 LOGICAL UNIT LISTING FOR  MSOS 5.0 TEST SYSTEM 5          09/23/76
LU. EQUIPMENT DESCRIPTION              READ/WRITE   CLASS CODE      EQ NO
01. SOFTWARE  CORE ALLOCATOR           READ/WRITE   NO CLASS CODE   EQ 00
02. SOFTWARE  DUMMY ALTERNATE DEVICE   READ/WRITE   NO CLASS CODE   EQ 00
03. SOFTWARE  DUMMY ALTERNATE DEVICE   READ/WRITE   NO CLASS CODE   EQ 00
04. 1810-1 LIAT CRT/PRINTER            READ/WRITE   TELETYPE        EQ 01
05. COSY UNIT                          READ/WRITE   MAGNETIC TAPE   EQ 00
06. 1860-92 LCTT 9TK MAG TAPE          READ/WRITE   MAGNETIC TAPE   EQ 09
07. PSEUDO TAPE UNIT                   READ/WRITE   MAGNETIC TAPE   EQ 00
08. 1833-1 STORAGE MODULE DRIVE 50MB   READ/WRITE   MASS STORAGE    EQ 14
09. 1827-30/60 LINE PRINTER            WRITE ONLY   LINE PRINTER    EQ 04
10. 1829-30/60 CARD READER             READ  ONLY   CARD RDR/PUNCH  EQ 11
11. 1860-92 LCTT 9TK MAG TAPE          READ/WRITE   MAGNETIC TAPE   EQ 09
12. 1827-30/60 LINE PRINTER            WRITE ONLY   LINE PRINTER    EQ 04
13. PSEUDO TAPE UNIT                   READ/WRITE   MAGNETIC TAPE   EQ 00
14. UNDEFINED UNIT TYPE                READ/WRITE   MASS STORAGE    EQ 00
15. 1832-5 CASSETTE MAG TAPE           READ/WRITE   MAGNETIC TAPE   EQ 07
16. 1860-92 LCTT 9TK MAG TAPE          READ/WRITE   MAGNETIC TAPE   EQ 09
17. 1860-72 LCTT 7TK MAG TAPE          READ/WRITE   MAGNETIC TAPE   EQ 09
18. PSEUDO DISK DRIVER                 READ/WRITE   MASS STORAGE    EQ 00
19. PSEUDO DISK DRIVER                 READ/WRITE   MASS STORAGE    EQ 00
20. PSEUDO DISK DRIVER                 READ/WRITE   MASS STORAGE    EQ 00
*U
```



($FFFE_{16}$ WORDS FOR A 1700 SERIES OR 18-20 COMPUTER WITHOUT EXTENDED MEMORY DRIVER. $F7FF_{16}$ WORDS FOR AN 18-20 COMPUTER WITH EXTENDED MEMORY DRIVER)

†OPTIONAL ITEMS.

REQUIREMENTS FOR N4, SIZE OF ALLOCATABLE AREA 4 M

The size of allocatable area 4 is N4. The value of N4 must be large enough so that the largest of the following programs can run in the area available at level 4:

1. Any system library programs that have a core request priority level of four. In a typical system this includes:

   ODEBUG

   System Checkout Package (SYSCOP)

   MIPRO

   File Manager

   Text Editor (EDITOR)

   Engineering File Modules

   SCMM (1700 computer only)

   Verification Programs

2. Any user programs that are to use this area

If the file manager is to be used by the background, N4 must be large enough so that even if the job processor and the protect processor occupy all of the areas available to levels 1 through 3, there is enough area left for the file manager to run. If the unbuffered protect processor, UNPROTP, is in the system and a file manager request is made from the background, a core swap cannot be made since background input/output would be in progress. Even if the buffered protect processor, BPROTP, is in the system, only a maximum of 96 words could be swapped.

The minimum allocatable core area needed by the file manager is:

$$P + I \cdot 96 + 118$$

Where: P is the size of the largest file manager request processor (store sequential, store direct, etc.).

I is 0 if there are no indexed files in the system.

I is 1 if there are indexed files in the system and the expected number of key values declared when a file is defined is less than 8465 for all files in the system.

I is 2 if there is at least one indexed file in the system for which the expected number of key values is declared at the time of file definition to be greater than or equal to 8465.

The constant value 118 is the sum of 96 words for the file information segment (FIS) directory, 16 words for one file information segment (FIS), and 6 words for the header appended to a FIS when a FIS is in core. The value $I \cdot 96$ is the size of the largest key information segment (KIS) directory for any file in the system. Thus, for background file manager requests to be possible, there must be:

$$N4 \geq P + I \cdot 96 + 118$$

If the file manager text editor is to be used, the lengths of the text editor modules are such that there must be:

$$N4 \geq 2048$$

Note that this is a minimum. If more core is available in area 4, the file manager may be able to have more than one processor as well as more than one KIS directory and/or FIS in core at once, thus increasing throughput. If the value of N4 is small, it may be wise to adjust the file manager timeout parameter so that core areas allocated for the file manager are released soon after they have been used. In a standard released system, N4 is set to the minimum. In a system that allows file manager requests by background programs, a released system allows at least 2048 words as the value of N4.

The skeleton may be obtained by using the skeleton editor. Mount the installation materials on the device from which they are to be read.

Enter the job processor.

*JOB when already in batch mode

The system outputs on the comment device:

J

NOTE

Each of the following entries are followed by a carriage return.

Enter:

*SKED

The system outputs:

SKED IN

NEXT

Enter:

ADF, lu, 1

Where: lu is the logical unit on which the installation materials are mounted. The installation file is the second file in the installation materials. (If installation materials are on cards and the first file (SLIP) has been removed, this step is omitted.)

The system outputs:

NEXT

Enter:

Build, lu

Where: lu is defined as above.

The system outputs:

ANY MORE INPUT

Enter:

CR

The system outputs:

NEXT

Enter:

DUMP, lu'

Where: lu' is the logical unit of the tape unit or card punch on which the skeleton is to be output.

The skeleton is output on the specified logical unit, and the skeleton records and corresponding numbers are listed on the list device.

The system outputs:

NEXT

Enter:

EXIT

Control is thus returned to the job processor.

*B SKELETON RECORD FORMAT

O

The basic format of a *B skeleton record is as follows:

```
*B  'aaaaaa'  'bbbbbb'
```

The *B must be in columns 1 and 2. Quote marks may begin anywhere after column 2. Embedded blanks are significant. The name of the program is specified by a one-to-six character name (aaaaaa), enclosed by single quote marks. The identification field (bbbbbb) provides the capability to differentiate between programs having the same name. A *B record in the skeleton directs LIBILD to retrieve a program module or file from the LIBILD input libraries and write the module or file to the installation file. Leaving the identification field blank (without even quote marks) causes the first copy of several copies or the only copy of a program to be retrieved.

When making additions to a standard system as described in section 5, the name field of each *B record must be as specified, but the identification field, including quotes, may be omitted. This is because each module is a new module and does not replace an existing module with the same name. If the identification field is included, those characters included in the field must be as specified in the relevant section of this handbook. Not all the characters in the identification field must be included, but a single quote must appear in the column immediately following the last character that is included. This is because blanks within the identification field are significant.

For example, if the handbook specifies:

```
*B 'ADDNOW' 'DECK-ID A52 MSOS 5.0 NOW'
```

the following would be acceptable:

```
*B 'ADDNOW' 'DECK ID A52 MSOS 5.0 NOW'
*B 'ADDNOW' 'DECK-ID'
*B 'ADDNOW'
```

The following would not be acceptable:

```
*B 'ADDNOW' 'DECK-ID  '
*B 'ADDNOW' 'DECK '
*B 'ADD'
```

The first two lines are not acceptable because trailing blanks are included within the single quotes.

Refer to appendix P for comments on skeleton modification.

When adding a product to a system, it is necessary to add a number of *B records to the system skeleton. One *B record for each module or file in a set of binaries may be generated by using the skeleton editor, SKED. Using SKED, the BUILD command is used to generate these *B records. When punched cards are not used in the system, this method is not very helpful. If the system does include punched card equipment, however, this method of generating *B records may save time.

Generation of *B records does not in itself constitute the required skeleton modifications for adding to a system. For example, if a *B record is generated for each module in the FORTRAN binaries file, these *B records are not in the order specified in section 6. To form a proper skeleton addition, the *B records must be duplicated and re-ordered as necessary. Control statements, other than *B records, must be added as needed.

VERIFICATION KEY
MONITOR TEST PROGRAM LOAD
*DM, *DL AND LULIST CONTROL RECORDS
FILE MANAGER AND PSEUDO TAPE TEST PROGRAM LOAD
JOB FILE TESTS AND CONTROL RECORDS
JOB FILE VERIFICATION DATA
MAGNETIC TAPE SIMULATOR TEST
ASSEMBLER, LIBILD, AND FORTRAN PROGRAM LOAD
ASSEMBLER SOURCE RECORDS
ASSEMBLER VERIFICATION DATA
LIBILD CONTROL RECORDS AND DATA
LIBILD VERIFICATION DATA
FORTRAN COMPILER SOURCE RECORDS
A COMPILER VERIFICATION DATA
B COMPILER VERIFICATION DATA
FORTRAN LIBRARY TESTS AND DATA
FORTRAN LIBRARY VERIFICATION DATA

DOUBLE-PRECISION LIBRARY TESTS AND DATA
DOUBLE-PRECISION VERIFICATION DATA
RE-ENTRANT FORTRAN PROGRAM LOAD
RE-ENTRANT FORTRAN TEST DATA
RE-ENTRANT FORTRAN VERIFICATION DATA
RE-ENTRANT DOUBLE-PRECISION FORTRAN PROGRAM LOAD
DOUBLE-PRECISION TEST DATA
DOUBLE-PRECISION VERIFICATION DATA
RPG II COMPILER SOURCE RECORDS
RPG II COMPILER VERIFICATION DATA
RPG II RUNTIME BINARIES AND AND DATA
FILE MARK
RPG II RUNTIME VERIFICATION DATA
SORT/MERGE SOURCE CONTROL CARDS AND DATA
FILE MARK
SORT/MERGE VERIFICATION DATA
VERIFICATION TEST EXECUTIVE PROGRAM LOAD
FILE MARK

Test Executive Errors:

*** ERROR IN TEST EXECUTIVE

Monitor Test Errors:

*** ERROR IN SCHEDULE REQUEST
*** ERROR IN TIMER REQUEST
*** ERROR IN SPACE REQUEST
*** ERROR IN DIRECTORY SCHEDULE
*** ERROR IN DISABLE-SCHEDULE
*** ERROR IN ENABLE-SCHEDULE
*** ERROR IN PARTITION SET-UP
*** ERROR IN PARTITIONED CORE REQUEST

File Manager Test Errors:

*** ERROR IN FILE DEFINITION
*** ERROR IN STORE SEQUENTIAL
*** ERROR IN RETRIEVE SEQUENTIAL
*** ERROR IN SEQUENTIAL FILE TEXT
*** ERROR IN LOCK FILE
*** ERROR IN UNLOCK FILE
*** ERROR IN STORE DIRECT
*** ERROR IN RETRIEVE DIRECT
*** ERROR IN DIRECT RETRIEVE TEXT
*** ERROR IN DEFINE INDEXED
*** ERROR IN STORE INDEXED
*** ERROR IN RETRIEVE INDEXED
*** ERROR IN INDEXED FILE TEXT
*** ERROR IN STORE INDEXED ORDERED
*** ERROR IN RETRIEVE INDEXED ORDERED
*** ERROR IN RELEASE FILE

Pseudo Tape Test Errors:

*** ERROR IN FWRITE REQUEST
*** ERROR IN FREAD REQUEST
*** ERROR IN WRITE REQUEST
*** ERROR IN READ REQUEST
*** ERROR IN RECORD TEXT
*** ERROR IN NUMBER OF FILE RECORDS
*** ERROR IN MOTION REQUEST
*** ERROR IN BACKSPACE RECORD PAST LOAD POINT
*** ERROR IN BACKSPACE FILE PAST LOAD POINT
*** ERROR IN BACKSPACE FROM LOADPOINT
*** ERROR IN READ RECORD PAST FILE MARK
*** ERROR IN BACKSPACE RECORD OVER FILE MARK
*** ERROR IN READ RECORD PAST END OF TAPE
*** ERROR IN ADVANCE RECORD OVER FILE MARK
*** ERROR IN JOB FILE LISTING
*** ERROR IN VERIFICATION TEST I/O

Magnetic Tape Simulator Test Errors:

*** ERROR IN FWRITE REQUEST
*** ERROR IN FREAD REQUEST
*** ERROR IN WRITE REQUEST
*** ERROR IN READ REQUEST
*** ERROR IN RECORD TEXT
*** ERROR IN MOTION REQUEST
*** ERROR IN BACKSPACE FROM LOADPOINT
*** ERROR IN READ RECORD PAST FILE MARK
*** ERROR IN BACKSPACE RECORD OVER FILE MARK
*** ERROR IN ADVANCE RECORD OVER FILE MARK
*** ERROR IN VERIFICATION TEST I/O

Macro Assembler Test Errors:

*** ERROR IN LISTING LINE NNNN
*** ERROR IN CROSS-REFERENCE
*** ERROR IN BINARY XXX BLOCK
*** ERROR IN VERIFICATION TEST I/O

Library Builder Test Errors:

*** ERROR IN LIBRARY BUILDER OUTPUT
*** ERROR IN VERIFICATION TEST I/O

FORTRAN Compiler Test Errors:

*** ERROR IN LISTING LINE NNNN
*** ERROR IN PROGRAM REFERENCE
*** ERROR IN BINARY XXX BLOCK
*** ERROR IN VERIFICATION TEST I/O

FORTRAN Library Test:

*** ERROR IN FORMATTED INPUT-OUTPUT
*** ERROR IN ENCODE-DECODE
*** ERROR IN ARITHMETIC LIBRARY
*** ERROR IN MONITOR INTERFACE
*** ERROR IN VERIFICATION TEST I/O

Double-Precision Test Errors:

*** ERROR IN FORMATTED INPUT-OUTPUT
*** ERROR IN ENCODE-DECODE
*** ERROR IN ARITHMETIC LIBRARY
*** ERROR IN VERIFICATION TEST I/O

Re-Entrant FORTRAN Library Test Errors:

*** ERROR IN MULTIPROGRAMMING EXECUTION
*** ERROR IN FORMATTED INPUT-OUTPUT
*** ERROR IN ARITHMETIC LIBRARY
*** ERROR IN MONITOR INTERFACE
*** ERROR IN VERIFICATION TEST I/O

Re-Entrant Double-Precision Test Errors:

*** ERROR IN FORMATTED INPUT-OUTPUT
*** ERROR IN ARITHMETIC LIBRARY
*** ERROR IN VERIFICATION I/O

RPG II Compiler Test Errors:

*** ERROR IN LISTING LINE NNNN
*** ERROR IN CROSS-REFERENCE
*** ERROR IN BINARY XXX BLOCK
*** ERROR IN VERIFICATION TEST I/O

RPG II Runtime Test Errors:

*** ERROR IN RPG RUNTIME
*** ERROR IN VERIFICATION TEST I/O

Sort/Merge Test Errors:

*** ERROR IN SORT/MERGE
*** ERROR IN VERIFICATION TEST I/O

```
          ASSEMD                          DATE: 11/11/11

0001                      NAM  ASSEMD
0002                  *   MSOS VERIFICATION TEST ASSEMBLER SOURCE PROGRAM
0003                  *   1700 MASS STORAGE OPERATING SYSTEM VERSION 4.1
0004                  *   SMALL COMPUTER DEVELOPMENT DIVISION, LA JOLLA, CALIFORNIA
0005                  *   COPYRIGHT CONTROL DATA CORPORATION 1975
0006                  *
0007                  *   THIS PROGRAM, WHEN CORRECTLY ASSEMBLED, VERIFIES THAT THE
0008                  *   MACRO ASSEMBLER IS FUNCTIONING PROPERLY.
0009                  *
0010                  *   THE FOLLOWING ASSEMBLER FEATURES ARE EXERCISED.
0011                  ****************************************************************
0012                  *
0013                  *     NOTE: NOT ALL ENHANCED INST. ARE TESTED BECAUSE OF POSSIBLE
0014                  *      SCRATCH OVERFLOW.
0015                  ****************************************************************
0016                  *
0017                  *   I MACHINE INSTRUCTIONS
0018                  *        1. STORAGE REFERENCE
0019                  *        2. REGISTER
0020                  *        3. SHIFT
0021                  *        4. SKIP
0022                  *        5. INTERREGISTER TRANSFER
0023                  *
0024                  *   II PSEUDO INSTRUCTIONS
0025                  *        1. SUBPROGRAM LINKAGE
0026                  *        2. DATA STORAGE
0027                  *        3. CONSTANT DECLARATION
0028                  *        4. ASSEMBLER CONTROL
0029                  *        5. LISTING CONTROL
0030                  *        6. MACRO DEFINITION
0031                  *
0032                  *   III STANDARD SYSTEM MACROS
0033                  *        1. SUBSET OF ABOVE
0034                  *
0035                  *   THIS PROGRAM IS DESIGNED TO BE
0036                  *   NON-EXECUTABLE ****************
0037                  *
0038                  *   VERIFICATION OF CORRECT ASSEMBLY OF ASSEMD IS DONE BY
0039                  *   THE ASSEMBLER VERIFICATION PROGRAM, ASSEMT
0040                  *   WHICH RUNS UNDER THE MSOS VERIFICATION TESTS.
0041                  *   BOTH LISTING AND BINARY OUTPUT OF ASSEMD ARE VERIFIED IN ASSEMT.
0042                  *
0043                  *   MACRO DEFINITION, LOC
0044                  *
0045                  XYZ MAC  P1,P2,P3,P4,P5
0046                      LOC  A
0047                      LDA  'P1'
0048                      'P2',,'P3'
0049                      S'P4'P 'A'-*-1
0050                      JMP','  'P5'
0051                  'A' ENA  1
0052                      EMC
0053                  *


0054                  *   MACRO IFC AND EIF
0055                  *
0056                  A   MAC  P1,P2,P3,P4
0057                  I1  IFC  *,EQ,'P1'
0058                      LDA  'P2'
0059                      EIF  I1
0060                  I2  IFC  *,NE,'P1'
0061                      LDA  'P3'
0062                      EIF  I2
0063                      STA  'P4'
0064                      EMC

0066                  *   EQUATE DESIGNATIONS

0068          0022        EQU  ZERO($22),AMONI($F4)
              00F4
0069          7FFE        EQU  COUNT(-1)
0070          0020        EQU  XFER($20)
0071          000F        EQU  PLACE($F),FIVE(5)
              0005

0073                  *   INTERNAL ENTRY POINT DESIGNATION

0075                      ENT  ASSEMD

0077 P0000 0B00   ASSEMD  NOP  0
```

```
0079                    *       STORAGE REFERENCE INSTRUCTIONS

0081 P0001 C000   X1    LDA   =N$F20E
     P0002 F20E
0082 P0003 E000   X2    LDQ   =XX1
     P0004 0001 P
0083                    LR1   X3+1
0083 P0005 0481
0083 P0006 C000
0083 P0007 0002
0084 P0008 8000   X3    ADD   =A00
     P0009 3030
0085                    ARQ*  X4
0085 P000A 0485
0085 P000B 8001
0086 P000C 90FF   X4    SUB-  $FF
0087                    SB3   X5
0087 P000D 0483
0087 P000E 9000
0087 P000F 0003
0088                    SB1*  X3
0088 P0010 0481
0088 P0011 90F6
0089 P0012 F4FF   X5    ADQ-  (I)
0090 P0013 A400   X6    AND+  $100
     P0014 0100
0091                    AN1*  X4
0091 P0015 0481
0091 P0016 A0F5
0092                    AM2   X7
0092 P0017 0482
0092 P0018 A100
0092 P0019 0003
0093                    OR3*  X7
0093 P001A 0483
0093 P001B D001
0094 P001C B400   X7    EOR*  ($100)
     P001D 8100
0095 P001E 281E   X8    MUI*  X24
0096 P001F 3CE8   X9    DVI*  (X3)
0097 P0020 6800   X10   STA   X60
     P0021 0043
0098 P0022 4C00   X11   STQ   (X60)
     P0023 0041
0099                    SR1*  (X12-5)
0099 P0024 04C1
0099 P0025 C1FD
0100                    SR4*  X12+1
0100 P0026 0484
0100 P0027 C102
0101 P0028 1A00   X12   JMP   X59+2,Q
     P0029 0039
0102 P002A 54F4   X13   RTJ-  (AMONI)


0103                    SJA   X2
0103 P002B 0486
0103 P002C 5000
0103 P002D FFD5
0104 P002E 0500   X14   RAQ+  $7ACE,I
     P002F 7ACE
0105                    DAP   4
0105 P0030 06C4
0106 P0031 7FD4   X15   SPA*  (X1+5),B
0107 P0032 C123   X16   LDA-  $23,I
0108                    LR3-  $22
0108 P0033 0403
0108 P0034 C022
0109 P0035 E223   X17   LDQ-  $23,Q
0110 P0036 8323   X18   ADD-  $23,B
0111 P0037 9622   X19   SUB-  (ZERO),Q
0112 P0038 F722   X20   ADQ-  (ZERO),B
0113 P0039 A9C9   X21   AND*  X2,I
0114 P003A B802   X22   EOR*  X24,B
0115 P003B 2DD6   X23   MUI*  (X5),I
0116 P003C 3E06   X24   DVI*  (X30),Q

0118                    *       REGISTER INSTRUCTIONS

0120 P003D 0000   X25   SLS   0
0121 P003E 0222   X26   INP   X59-*
0122 P003F 03FE   X27   OUT   -1
0123 P0040 0A0A   X28   ENA   10
0124 P0041 0CCF   X29   ENQ   -$30
0125 P0042 09FE   X30   INA   COUNT
0126 P0043 0D40   X31   INQ   $40
0127 P0044 0B00   X32   NOP   0
0128 P0045 0400   X33   EIN   0
0129 P0046 0500   X34   IIN   0
0130                    DMI
0130 P0047 0B06
0131 P0048 0E00   X35   EXI   0
0132 P0049 0600   X36   SPB   0
0133 P004A 0700   X37   CPB   0
```

```
0135                    *       SHIFT INSTRUCTIONS                                      0190
                                                                                        0191
0137 P004B 0F41   X38   ARS   1                                                        0192
0138 P004C 0F2D   X39   QRS   13                                                       0193
0139 P004D 0F7F   X40   LRS   XFER-1                                                   0194
0140 P004E 0FC8   X41   ALS   8                                                        0195
0141 P004F 0FAF   X42   QLS   $F                                                       0196
0142 P0050 0FF9   X43   LLS   25                                                       0197
                                                                                        0198
0144                    *       SKIP INSTRUCTIONS                                       0199
                                                                                        0230
0146 P0051 010E   X44   SAZ   X59                                                      0201
0147 P0052 011E   X45   SAN   $F-1                                                     0202
                                                                                      0203
                                                                                        0204
                                                                                        0205
0148 P0053 012D   X46   SAP   $D                                                       0206
0149 P0054 013C   X47   SAM   X59-*                                                    0207
0150 P0055 014B   X48   SQZ   $B                                                       0208
0151 P0056 015A   X49   SQN   PLACE-FIVE                                               0209
0152 P0057 016C   X50   SQP   X60-*-1                                                 0210
0153 P0058 0178   X51   SQM   X58-X50                                                 0211
0154 P0059 0187   X52   S=S   X53+7                                                   0212
0155 P005A 0199   X53   SWN   X60                                                    0213
0156 P005B 01A5   X54   SOV   -5+10                                                  0214
0157 P005C 01B4   X55   SNO   $4                                                     0215
0158 P005D 01C3   X56   SPE   3                                                       0216
0159 P005E 01D2   X57   SNP   X59+1                                                   0217
0160 P005F 01E4   X58   SPF   X60                                                    0218
0161 P0060 01F0   X59   SNF   0                                                       0219
0162                    S1M   X61-*-1                                                 0220
0162 P0061 0073                                                                       0221
0163                    S2P   X67-*-1                                                 0222
0163 P0062 00A8                                                                       0223
0164                    S4Z   X63-*-1                                                 0224
0164 P0063 0003                                                                       0225
                                                                                        0226
0166                    *       INTERREGISTER TRANSFER INSTRUCTIONS                     0227
                                                                                        0228
0168 P0064 0804   X60   SET   A                                                       0229
0169 P0065 0842   X61   CLR   Q                                                       0230
0170 P0066 0821   X62   TRA   M                                                       0231
0171 P0067 080E   X63   TRM   A+Q                                                     0232
0172 P0068 0815   X64   TRQ   A+M                                                     0233
0173 P0069 0818   X65   TRB   Q+M                                                     0234
0174 P006A 0867   X66   TCA   A+Q+M                                                   0235
0175 P006B 084F   X67   TCM   A+M+Q                                                   0236
0176 P006C 0856   X68   TCQ   Q+A                                                     0237
0177 P006D 085D   X69   TCB   M+A                                                     0238
0178 P006E 0828   X70   AAM   M+Q                                                     0239
0179 P006F 0830   X71   AAQ   0                                                       0240
0180 P0070 083F   X72   AAB   Q+M+A                                                   0241
0181 P0071 086F   X73   EAM   Q+A+M                                                   0242
0182 P0072 0877   X74   EAQ   M+A+Q                                                   0243
0183 P0073 087F   X75   EAB   M+Q+A                                                   0244
0184 P0074 08AC   X76   LAM   A                                                       0245
0185 P0075 08B2   X77   LAQ   Q                                                       0246
0186 P0076 08B9   X78   LAB   M                                                       0247
0187 P0077 08E8   X79   CAM   0                                                       0248
0188 P0078 08F6   X80   CAQ   A+Q                                                     0249
0189 P0079 08FF   X81   CAB   M+A+Q                                                   0250
0190                    GPE                                                           0251
0190 P007A 0B08                                                                       0252
0191                    LUB   Q                                                       0253
0191 P007B 0BA0                                                                       0254
                                                                                      0255
                                                                                        0256
                                                                                        0257
0193                    *       EXTERNAL ENTRY POINTS                                   0258
                                                                                        0259
0195                            EXT   LOG1+LOG1A+INPTV4                                 0260
0196                            EXT*  DGNTAB                                           0261
0197 P007C C600 X               LDA*  LOG1A+Q                                          0262
     P007D 7FFF X                                                                       0263
0198 P007E B900 X               EOR   DGNTAB+I                                         0264
     P007F 7FFF X                                                                       0265
                                                                                        0266
0200                    *       COMMON DESIGNATIONS                                     0267
                                                                                        0268
0202          0000 C COMSTR COM   BLOCK(10),BLOCK1(0),BLOCK2+BLOCK3(FIVE)             0269
              000A C                                                                    0270
              000A C                                                                    0271
              000B C                                                                    0272
                                                                                        0273
0204                    *       BSS AND BZS ASSIGNMENTS                                 0274
                                                                                        0275
0206 P0080 0002   BSSSTR BSS   BLOCK4(2),BLOCK5(0),BLOCK6                              0276
     P0082 0000                                                                         0277
     P0082 0001                                                                         0278
0207 P0083 0005            BSS   BLOCK7(FIVE)                                         0279
0208 P0088 0004   BZSSTR BZS   BLOCK8(4)                                             0280
0209 P008C 0000            BZS   BLOCK9(0),BLOCKA                                     0281
     P008C 0001                                                                         0282
0210 P008D 0005            BZS   BLOCKB(FIVE)                                         0283
```

```
0212                    *       ADC,ALF,NUM,VFD,STANDARD SYSTEM MACRO,COMMENTS

0214 P0092 54F4                 RTJ- (AMONI)            DO A MONITOR ROST
0215 P0093 0901                 VFD  N1/0,N1/0,N5/54,N1/1,N4/0,X4/5-4  FREAD REL
0216 P0094 0014                 ADC* COMP1+1            COMPLETION
0217 P0095 0000                 ADC  0                  THREAD
0218 P0096 18FC                 ADC  $18FC              V,M,A,LU
0219 P0097 000C                 NUM  12                 LENGTH
0220 P0098 0008                 ADC  ALFLOC-**5         BUFFER FWA
0221                            EXIT                    SUFFICIENT TO EXERCISE SYSTEM MACROS
0221 P0099 54F4
0221 P009A 0A00
0222 P009B 4D45   ALFLOC ALF  $,MESSAGE IN ASCII FORMAS
     P009C 5353
     P009D 4147
     P009E 4520
     P009F 494E
     P00A0 2041
     P00A1 5343
     P00A2 4949
     P00A3 2046
     P00A4 4F52
     P00A5 4D41
0223 P00A6 5420                 VFD  A16/T
0224 P00A7 0161   COMP1  SQP  1                 COMPLETION

0226                    *       DEC ASSIGNMENTS

0228 P00A9 00E0   DECASN DEC  35D-186
0229 P00A9 F73F   DECAS1 DEC  -3586
0230 P00AA 07FF                 DEC  32760B-4
0231 P00AB 29EE   DECAS2 DEC  32761D-5B15,+625D-2B3
     P00AC 0032
0232 P00AD 2710   DECAS3 DEC  10D3

0234                    *       LISTING CONTROL

0240                    *       THIS LINE SHOULD LIST

0243                    *       USER DEFINED MACRO USAGE

0245                    TAG1    XYZ* BLOCK1,STA,*BLOCK2,I*,
0246                            XYZ* Q,LABEL1
0246 P00B0 C400
     P00B1 000A C
0246 P00B2 6500
     P00B3 000A C
0246 P00B4 0161
0246 P00B5 1802
0246 P00B6 0A01
0247 P00B7 0B00   LABEL1 NOP  0

0249                            A    *,BLOCK3,BLOCK4,BLOCK5
0249 P00B8 C400
     P00B9 000B C
0249 P00BA 6800
     P00BB FFC6

0251                    *       DATA AND ORG

0253          0000 D DATSTR DAT  DATA1(5),DATA2(0),DATA3,DATA4(FIVE)
              0005 D
              0005 D
              0006 D
0254          0000 D          ORG  DATA1
0255 D0000 7FFF                 ADC  $7FFF,$0F0F,$70FF,$7ACE,$3030
     D0001 0F0F
     D0002 70FF
     D0003 7ACE
     D0004 3030
0256          00BC P          ORG*
0257 P00BC 0B00   ENDDAT NOP  0                 END OF DATA
0258          0005 D          ORG  DATA2
0259 D0005 000A                 NUM  $000A,$000B,$000C,$000D,$000E,$000F
     D0006 000B
     D0007 000C
     D0008 000D
     D0009 000E
     D000A 000F
0260          00BD P          ORG*

0262                    *       IFA AND EIF
```

```
0264       0000          EQU  ASMFLG(0),ASMFL1(1)
           0001
0265                NAM3  IFA  ASMFLG.FQ.0
0266 P00BD 0101     OP1   SAZ  1
0267                      EIF  NAM3
0268                      IFA  ASMFLG.GT.ASMFL1
0269                OP2   SAZ  2
0270                      EIF

0272                      END  ASSEMD

PGM=  00BE (   190)   COM = 0010 (    16)   DAT = 000B (    11)
```

EQUIVALENCES

DEF.LINE	NAME	VALUE		REFERENCED AT LINE NUMBER
0000	I	00FF	(000255)	0099
0069	ZERO	0022	(000034)	0111, 0112
0068	AMONI	00F4	(000244)	0102, 0214
0069	COUNT	7FFE	(032766)	0125
0070	XFER	0020	(000032)	0139
0071	PLACE	000F	(000015)	0151
0071	FIVE	0005	(000005)	0151, 0202, 0207, 0210, 0253
0264	ASMFLG	0000	(000000)	0265, 0268
0264	ASMFL1	0001	(000001)	0268

SYMBOLS

DEF.LINE	NAME	ADDRESS	REFERENCED AT LINE NUMBER
0075	ASSEMD	0000	0075
0081	X1	0001	0082, 0106
0082	X2	0003	0103, 0113
0084	X3	0008	0083, 0088, 0088, 0088, 0096
0086	X4	000C	0085, 0085, 0085, 0091, 0091, 0091
0089	X5	0012	0087, 0115
0090	X6	0013	
0094	X7	001C	0092, 0093, 0093, 0093
0095	X8	001E	
0096	X9	001F	
0097	X10	0020	
0098	X11	0022	
0101	X12	0023	0099, 0099, 0099, 0100, 0100, 0100
0102	X13	002A	
0104	X14	002E	
0106	X15	0031	
0107	X16	0032	
0109	X17	0035	
0110	X18	0036	
0111	X19	0037	
0112	X20	0038	
0113	X21	0039	
0114	X22	003A	
0115	X23	003B	
0116	X24	003C	0095, 0114
0120	X25	003D	
0121	X26	003E	
0122	X27	003F	
0123	X28	0040	
0124	X29	0041	
0125	X30	0042	0116
0126	X31	0043	
0127	X32	0044	
0128	X33	0045	
0129	X34	0046	
0131	X35	0048	
0132	X36	0049	
0133	X37	004A	
0137	X38	0048	
0138	X39	004C	
0139	X40	004D	

```
0140    X41     004E
0141    X42     004F
0142    X43     0050
0144    X44     0051
0147    X45     0052
0148    X46     0053
0149    X47     0054
0150    X48     0055
0151    X49     0056
0152    X50     0057            0153
0153    X51     0058
0154    X52     0059
0155    X53     005A            0154
0156    X54     0058
0157    X55     005C
0158    X56     005D
0159    X57     005E
0160    X58     005F            0153
0161    X59     0060            0101. 0121. 0146. 0149. 0159
0168    X60     0064            0097. 0098. 0152. 0155. 0160
0169    X61     0065            0162. 0162. 0162
0170    X62     0066
0171    X63     0067            0164. 0164. 0164
0172    X64     0068
0173    X65     0069
0174    X66     006A
0175    X67     0068            0163. 0163. 0163
0176    X68     006C
0177    X69     006D
0178    X70     006E
0179    X71     006F
0180    X72     0070
0181    X73     0071
0182    X74     0072
0183    X75     0073
0184    X76     0074
0185    X77     0075
0186    X78     0076
0187    X79     0077
0188    X80     0078
0189    X81     0079
0202    BLOCK   0000
0202    BLOCK1  000A            0246
0202    BLOCK2  000A            0246
0202    BLOCK3  0008            0249
0206    BLOCK4  0080
0206    BLOCK5  0082            0249
0206    BLOCK6  0082
0207    BLOCK7  0083
020A    BLOCK8  008A
0209    BLOCK9  008C
0209    BLOCKA  008C
0210    BLOCKB  008D


0222    ALFLOC  009B            0220
0224    COMP1   00A7            0216
0228    DECASN  00A8
0229    DECAS1  00A9
0231    DECAS2  00AB
0232    DECAS3  00AD
0245    TAG1    0090
0244    IOD     0086            0246
0247    LABEL1  0097            0246
0253    DATA1   0000            0254
0253    DATA2   0005            0258
0253    DATA3   0005
0253    DATA4   0006
0257    ENDDAT  008C
0266    OP1     008D
```

EXTERNALS

```
DEF.LINE  NAME    VALUE         REFERENCED AT LINE NUMBER

0195      LOG1    7FFF
0195      LOG1A   007D          0197
0195      INPTV4  00AF          0238
0196      DGNTAB  007F          0198
```

*** ALPHABETICAL SORT OF SYMBOLS ***

ALFLOC	0222	AMONI	0068	ASMFL1	0264	ASMFLG	0264	ASSEMD	0075	BLOCK	0202	BLOCK1	0202	BLOCK2	0202	BLOCK3	0202
BLOCK4	0205	BLOCK5	0206	BLOCK6	0206	BLOCK7	0207	BLOCK8	020A	BLOCK9	0209	BLOCKA	0209	BLOCKB	0210	COMP1	0224
COUNT	0069	DATA1	0253	DATA2	0253	DATA3	0253	DATA4	0253	DECAS1	0229	DECAS2	0231	DECAS3	0232	DECASN	0228
DONTAR	0196	ENDDAT	0257	FIVE	0071	I	0000	INPTV4	0195	LABEL1	0247	LOG1	0195	LOG1A	0195	OP1	0246
PLACE	0071	TAG1	0245	X1	0061	X10	0097	X11	0098	X12	0101	X13	0102	X14	0104	X15	0106
X16	0107	X17	0109	X18	0110	X19	0111	X2	0082	X20	0112	X21	0113	X22	0114	X23	0115
X24	0116	X25	0120	X26	0121	X27	0122	X28	0123	X29	0124	X3	0084	X30	0125	X31	0126
X32	0127	X33	0128	X34	0129	X35	0131	X36	0132	X37	0133	X38	0137	X39	0138	X4	0086
X40	0139	X41	0140	X42	0141	X43	0142	X44	0146	X45	0147	X46	0148	X47	0149	X48	0150
X49	0151	X5	0089	X50	0152	X51	0153	X52	0154	X53	0155	X54	0156	X55	0157	X56	0158
X57	0159	X58	0160	X59	0161	X6	0090	X60	0168	X61	0169	X62	0170	X63	0171	X64	0172
X65	0173	X66	0174	X67	0175	X68	0176	X69	0177	X7	0094	X70	0178	X71	0179	X72	0180
X73	0181	X74	0182	X75	0183	X76	0184	X77	0185	X78	0186	X79	0187	X8	0095	X80	0188
X81	0189	X9	0096	XFER	0070	ZERO	0068	[00	0246								

```
*JOB
*FTN
 OPT LXCO
      PROGRAM FTNMAY
C     MSOS VERIFICATION TEST COMPILER SOURCE PROGRAM 1
C     1700 MASS STORAGE OPERATING SYSTEM VERSION 4.1
C     SMALL COMPUTER DEVELOPMENT DIVISION, LA JOLLA, CALIFORNIA
C     COPYRIGHT CONTROL DATA CORPORATION 1975
C
C     THIS PROGRAM IS DESIGNED TO EXERCISE
C     THE FORTRAN COMPILER AND IS NON-EXECUTABLE
C     FTNTST VERIFIES LIST AND BINARY OUTPUT OF FTNMAY COMPILATION
C
C
C     EXTERNAL STATEMENT
C
      EXTERNAL FTNSUB,FTNFCN
C
C     RELATIVE STATEMENT
C
      RELATIVE REL1,REL2,REL3,BLKDAT
C
C     TYPE STATEMENT
C
      INTEGER AINT1,BINT2,CINT3,GINT7(5),HINT8,
     1IINT9,JINT10,FINT6
      REAL  AREAL1,BREAL2,CREAL3(3,3),IREAL1,JREAL2,
     1KREAL3(9)
0     DOUBLE PRECISION MDBL1,NDBL2,ODBL3,PDBL4(3,5),
     1QDBL5,RDBL6,SDBL7,TDBL8(2,4),UDBL9,VDBL10,WDBL11
      SINGLE DINT4,EINT5,KINT11
C
C     DIMENSION STATEMENT
C
      DIMENSION LINT12(5),QDBL5(2,3,4),AINT1(10),
     1AREAL1(3,4),FINT6(5)
C
C     BYTE AND SIGNED BYTE STATEMENT
C
      BYTE (FINT6,LINT12(1)(13=6))
      SIGNED BYTE (HINT8,GINT7(3)(7=0))
C
C      COMMON STATEMENT
C
      COMMON /LABEL/AREAL1,EINT5,MINT13(12),P4BL4
      COMMON /LABEL/HREAL8,WDBL11
      COMMON //BREAL2(2,2,2),BINT2,QDBL5
      COMMON AINT1,VDBL10(10),EREAL5
C
C     DATA STATEMENT
C
0     DATA  (GINT7(J),J=1,5)/$FDB9,$DB97,$B975,$9753,$7531/
      DATA ((CREAL3(I,J),I=1,3),J=1,3)/3567.508,1.2,
     15286.3254,98.6,110.9,0.000050,.1,1056.3219,36500000.0/
      DATA MDBL1/345.67D-03/,NDBL2/.34567D+5/,
     1ODBL3/34567.D-05/
C
C     EQUIVALENCE STATEMENT
C
      EQUIVALENCE (DREAL4,KREAL3(8))
      EQUIVALENCE (AINT1(6),LINT12(1)),(MDBL1,NDBL2)
C
C     STATEMENT FUNCTION,INTRINSIC FUNCTION,EXTERNAL
C     FUNCTION,FUNCTION SUBPROGRAM CALL
C
      MYFUNC (I,J,DREAL4,EREAL5,RDBL6) =
     1DFLT(ABS(I))*RDBL6/(SIN(EREAL5))**2
     2*ALOG(DREAL4)-SQRT(J)*FTNFCN(DREAL4,EREAL5,FREAL6,BINT2,CINT3,
     3AINT1(5))
C
C     ASSIGNMENT STATEMENT
C
10    I = LINT12(1)*LINT12(2)-LINT12(3)*BREAL2(1,2,1)/.005
     1**2
20    WDBL11 = (TDBL8(2,2)*(I/5)+365.568)/LINT12(3)**2+MYFUNC(I,
     1J,DREAL4,EREAL5,RDBL6)
C
C     LOGICAL IF,RELATIONAL EXPRESSIONS,UNCONDITIONAL GO TO,
C     LABLED ASSIGNMENT,SUBROUTINE CALL,FORMATTED WRITE,STOP,
C     LOGICAL EXPRESSIONS,PAUSE
C
30    IF (I.EQ.LINT12(4)) GO TO 40
```

```
40     IF (I.NE.LINT12(4)) J = I+1
50     IF (LINT12(5).GT.J) ASSIGN 800 TO IFORM
60     IF (J.GE.I) CALL FTNSUB(25,AREAL1,MDBL1)
70     IF (J.LT.I) WRITE (4,600)
80     IF (3.LE.AREAL1) STOP 6
90     IF (.NOT.(I.EQ.LINT12(4)).AND.(LINT12(5).GT.J).OR.
      1(J.LT.3)) PAUSE 7
C
C      FORMAT STATEMENT
C
500    FORMAT (//5F10.5,E10.2/15D11.7,3(I10,$4),2Z3,2A2,R1/)
600    FORMAT (1H0,22HREPLACE THIS STATEMENT,5X,
      1*COMMENT '1'*,'COMMENT *2**)
700    FORMAT (I8)
800    FORMAT (1H1,F6.4)
900    FORMAT (/D17.10)
C
C      RELEASE STATEMENT
C
       CALL RELESE (FTNMAY)
C
C       END STATEMENT
C
       END
 OPT LXARCOV
       SUBROUTINE FTNSUB (I,LREAL4,XDBL12)
C      MSOS VERIFICATION TEST COMPILER SOURCE PROGRAM 2
C      1700 MASS STORAGE OPERATING SYSTEM VERSION 4.1
C      SMALL COMPUTER DEVELOPMENT DIVISION, LA JOLLA, CALIFORNIA
C      COPYRIGHT CONTROL DATA CORPORATION 1975
C
C      THIS NONEXECUTABLE SUBPGM. IS DESIGNED TO EXERCISE THE COMPILER
C      FTNTST VERIFIES LIST AND BINARY OUTPUT OF FTNSUB COMPILATION
       REAL IREAL4
       DOUBLE PRECISION XDBL12
       SINGLE I,IBUF(58),IDAT(3),INUM,ITEMP(8)
       DATA (IDAT(I),I =1,3)/$0023,$FFFE,$001A/,INUM/5/
C
C      SETBFR,FORMATTED WRITE,IOERR,IRWERR
C
1      CALL SETBFR (IBUF,58)
0      WRITE (I,100)(IDAT(I),I=1,3),INUM
       IF (IOERR(0).EQ.-1) GO TO 50
       JERROR = IRWERR(0)
C
C      CHARACTER CONVERSION
C
50     CALL HEXASC(I,IBUF(1))
55     CALL HEXDEC(I,IBUF(1))
60     CALL ASCII(IBUF(1),I)
65     CALL DECHEX(IBUF(1),I)
70     CALL AFORM(IBUF(1),IDAT)
       CALL RFORM(IBUF(1),IDAT)
       CALL FLOATG(LREAL4,IBUF(1))
C
C      INPUT/OUTPUT
C
       CALL OUTINS(IDAT)
       CALL INPINS(IDAT)
       CALL ICONCT(IDAT)
       CALL OCONCT(IDAT)
C
C      FORTRAN/MONITOR INTERFACE
C
       IFLAG = $0011
       ASSIGN 75 TO ICOMP
       CALL FWRITE ($18F8,IBUF(1),40,ICOMP,IFLAG,ITEMP(1))
       CALL DISPAT
75     CALL SCHEDL (80,$1,I,ITEMP)
       CALL DISPAT
       CALL TIMER (1,$21,5,ITEMP)
       CALL DISPAT
80     N = LINK(0)
       K = ICLOCK(0)
C
C      ENCODE/DECODE
C
       ASSIGN 99 TO IFORM
       CALL ENCODE (IBUF,IFORM,3,IDAT)
       IFLAG = DECODE (IBUF,IFORM,3,IDAT)
       RETURN
99     FORMAT (I3)
100    FORMAT (/3I2,10H TERMINAL ,I2,11H TERMINATED)
       END
 OPT LXMVC
       REAL FUNCTION FTNFCN (A1,A2,A3,I1,I2,I3)
C      MSOS VERIFICATION TEST COMPILER SOURCE PROGRAM 3
C      1700 MASS STORAGE OPERATING SYSTEM VERSION 4.1
C      SMALL COMPUTER DEVELOPMENT DIVISION, LA JOLLA, CALIFORNIA
C      COPYRIGHT CONTROL DATA CORPORATION 1975
C
```

```
C     THIS NONEXECUTABLE SUBPGM. IS DESIGNED TO EXERCISE THE COMPILER
C     FTNTST VERIFIES LIST AND BINARY OUTPUT OF FTNFCN COMPILATION
      SINGLE LENGTH,BUFFER(50),ITEMP
C
C     ASSEMBLY CODE,CONTINUE STATEMENT
C
      ASSEM .15,$C8FE,$6400,*I1,$6400,ITEMP
      ASSEM .16,$54F4,*,$0901,*17,$0,$08F9,*(LENGTH),*BUFFER(1)
17    CONTINUE
C
C     ASSIGNED GO TO,COMPUTED GO TO
C
      GO TO I3,(20,30,40,50,60)
      GO TO (20,30,40,50,60),I2
C
C     ARITHMETIC IF
C
20    IF (A1/A2) 30,40,50
C
C     PAUSE STATEMENT
C
30    PAUSE 30
C
C     DO LOOP,UNFORMATTED READ,UNFORMATTED WRITE
C
40    DO 45 J =1,50,1
      READ (1) (BUFFER(I), I =1,50)
      WRITE (3) (BUFFER(I), I =1,50,1)
45    CONTINUE
50    DO 59 M =I2,1,-1
      DO 58 N =1,20,5
      A3 = FLOAT(M+N)+A3
58    CONTINUE
59    CONTINUE
C
C     OPEN MASS STORAGE FILE
C
60    OPEN 2,1,200,8,1
C
C     FORMATTED READ AND WRITE STATEMENTS
C
70    READ (6,200) (BUFFER(I),I=1,50)
      WRITE (8,200) (BUFFER(I),I=1,50)
C
C     TAPE CONTROL,BACKSPACE,ENDFILE,REWIND
C
      BACKSPACE 6
      ENDFILE 6
      REWIND 6
C
C     FUNCTION VALUE RETURN
C
      FTNFCN = A3+A1/FLOAT(BUFFER(20))
      RETURN
C
200   FORMAT (50(1X,I2))
      END
 OPT LXKVCO
      BLOCK DATA
C     MSOS VERIFICATION TEST COMPILER SOURCE PROGRAM 4
C     1700 MASS STORAGE OPERATING SYSTEM VERSION 4.1
C     SMALL COMPUTER DEVELOPMENT DIVISION, LA JOLLA, CALIFORNIA
C     COPYRIGHT CONTROL DATA CORPORATION 1975
C
C     THIS NONEXECUTABLE SUBPGM. IS DESIGNED TO EXERCISE THE COMPILER
C     FTNTST VERIFIES LIST AND BINARY OUTPUT OF BLOCK DATA COMPILATION
0     COMMON /ENTER/A,C,D,I,K
      DIMENSION A(4),B(4),C(5),D(2),I(3),J(3),K(2)
      EQUIVALENCE (A,B),(I,J)
      DATA A(1),A(2),A(3),A(4)/1.1,2.2,3.3,4.4/,C(1),C(2),C(3),C(4),C(5)
     */1.1,2.2,3.3,4.4,5.5/,D(1),D(2)/10.1,10.2/,I(1),I(2),I(3),K(1),
     *K(2)/1,2,3,4,5/
      END
 MON
*U
```

```
RPG II COMPILER                                                            PAGE 0001

        H                                                                 BCHART
        F*****************************************************************
        F*                                                               *
        F*  FIRST COMPILED 08/07/76.                                     *
        F*  GIVEN STATISTICS LISTED BELOW. JOB PREPARES BIORHYTHM CHARTS FOR  *
        F*  ANY NUMBER OF PERSONS WITH ACCOMPANYING DOCUMENTATION FOR FACH.   *
        F*                                                               *
        F*  PREPARE INPUT CARDS FOR EACH PERSON TO BE CHARTED AS FOLLOWS....  *
        F*                                                               *
        F*  CARD COLUMNS DESCRIPTION                                     *
        F*                                                               *
        F*    01 - 32     NAME OF PERSON TO BE CHARTED.                  *
        F*    33 - 40     DATA OF BIRTH  MMDDYYYY.                       *
        F*    41 - 46     CHART BEGINNING MONTH AND YEAR  MMYYYY.        *
        F*    47 - 50     NUMBFR OF MONTHS TO BE CHARTED NNNN.           *
        F*                                                               *
        F*****************************************************************
        F*
        F*
        F*
        FDATACARDIP  F  80  80    2       MFCU1
        FREPORT  O   F  96  96    2       PRINTER
        E                       MON     12  12  4
        E                       MCNT    12  12  2 0
        E                       PT      20  23  4
        E                       ET      20  28  4
        E                       IT      20  33  4
        E                       DOC      1  79 76
        E                       LIN         62  1
        E                       PHY         31  4
        E                       EMT         31  4
        E                       INT         31  4
        E                       CD           8  2 0
        E                       TYPE         8  3
        IDATACARDAA  01
        I                                         1  32 NAME
        I                                        33  340BMONTH
        I                                        33  340B
        I                                        35  360BDAY
        I                                        39  400BLY           88
        I                                        37  400BYEAR
        I                                        41  420CMONTH
RPG II COMPILER                                                            PAGE 0002

        I                                        41  420C
        I                                        45  460CLY           89
        I                                        43  460CYEAR
        I                                        47  500XMONTH
        C         N01           GOTO END
        C                       EXSR DOCSR
        C                       EXSR CHRTSR
        C             END       TAG
        CSR           DOCSR     BEGSR
        CSR                     SETOF                  30
        CSR                     Z-ADD1         YZ     20
        CSR           DOC#1     TAG
        CSR                     MOVELDOC,YZ    LINE   75
        CSR                     MOVE DOC,YZ    ZY     10
        CSR           ZY        COMP 1                     02
        CSR           ZY        COMP 2                     03
        CSR           ZY        COMP 3                     04
        CSR           ZY        COMP 4                     05
        CSR                     EXCPT
        CSR           YZ        ADD  1         YZ
        CSR           YZ        COMP 80                   06
        CSR         06          GOTO DOC#1
        CSR                     SETOF                  020304
        CSR                     SFTOF                  05
        CSR                     ENDSR
        CSR           CHRTSR    BEGSR
        CSR                     Z-ADD0         TMONTH 40
        CSR                     Z-ADD0         WORK   50
        CSR                     SETOF                  070809
        CSR                     SETOF                  109294
        CSRN88        BLY       DIV  4         TEST   40
        CSRN88                  MVR            LEAP   40   94
        CSR 88        BYEAR     DIV  400       TEST
        CSR 88                  MVR            LEAP        94
        CSR 94        BMONTH    COMP 2                     92
        CSR           BYEAR     COMP CYEAR             090710
        CSR 10        BMONTH    COMP CMONTH            090810
        CSR 07                  GOTO TAG01
        CSR 08                  GOTO TAG01
        CSR 09                  GOTO TAG23
        CSR 10                  GOTO TAG08
        CSR           TAG01     TAG
RPG II COMPILER                                                            PAGE 0003
```

```
0085      CSR                 MOVE BMONTH    M       20
0086      CSR       MCNT,M    SUB  BDAY      WORK
0087      CSR       WORK      ADD  1         WORK
0088      CSR 92    WORK      ADD  1         WORK
0089      CSR       TAG02     TAG
0090      CSR       M         COMP 12                    81
0091      CSR 81              GOTO TAG03
0092      CSR       M         ADD  1         M
0093      CSR 94    M         COMP 2                     92
0094      CSR 08    M         COMP CMONTH                85
0095      CSR 08 85 WORK      ADD  1         WORK
0096      CSR 08 85           GOTO TAG08
0097      CSR       WORK      ADD  MCNT,M    WORK
0098      CSR 92    WORK      ADD  1         WORK
0099      CSR                 GOTO TAG02
0100      CSR       TAG03     TAG
0101      CSR                 Z-ADDBYEAR     TYEAR   40
0102      CSR                 SETOF                      9294
0103      CSR       TAG04     TAG
0104      CSR       TYEAR     ADD  1         TYEAR
0105      CSR                 MOVE TYEAR     TLY     20
0106      CSR       TLY       COMP 00                    96
0107      CSRN96    TYEAR     DIV  4         TEST
0108      CSRN96              MVR            LEAP        94
0109      CSR 96    TYEAR     DIV  400       TEST
0110      CSR 96              MVR            LEAP        94
0111      CSR       TYEAR     COMP CYEAR                 95
0112      CSR 95              GOTO TAG05
0113      CSR       WORK      ADD  365       WORK
0114      CSR 94    WORK      ADD  1         WORK
0115      CSR                 GOTO TAG04
0116      CSR       TAG05     TAG
0117      CSR                 Z-ADD0         M
0118      CSR       TAG06     TAG
0119      CSR       M         ADD  1         M
0120      CSR 94    M         COMP 2                     92
0121      CSR       M         COMP CMONTH                82
0122      CSR 82              GOTO TAG07
0123      CSR       WORK      ADD  MCNT,M    WORK
0124      CSR 92 94 WORK      ADD  1         WORK
0125      CSR                 GOTO TAG06
0126      CSR       TAG07     TAG
 RPG II COMPILER                                                    PAGE 0004

0127      CSR       WORK      ADD  1         WORK
0128      CSR       TAG08     TAG
0129      CSRN92              MOVE MCNT,C    METEST  20
0130      CSR 92              MOVE 29        METEST
0131      CSR                 MOVE *  *      PHY
0132      CSR                 MOVE *  *      EMT
0133      CSR                 MOVE *  *      INT
0134      CSRN10    WORK      DIV  23        HASH    50
0135      CSRN10              MVR            P       20  23
0136      CSRN10 23           Z-ADD23        P           23
0137      CSRN10              Z-ADD0         D       20
0138      CSR 10              Z-ADD1         P
0139      CSR 10    BDAY      SUB  1         D
0140      CSR       TAG09     TAG
0141      CSR       D         ADD  1         D
0142      CSR                 MOVE PT,P      PHY,D
0143      CSR       D         COMP METEST                71
0144      CSR 71              GOTO TAG10
0145      CSR       P         ADD  1         P
0146      CSR       P         COMP 23                24
0147      CSR 24    P         SUB  23        P
0148      CSR                 GOTO TAG09
0149      CSR       TAG10     TAG
0150      CSRN10    WORK      DIV  28        HASH
0151      CSRN10              MVR            E       20  28
0152      CSRN10 28           Z-ADD28        E           28
0153      CSRN10              Z-ADD0         D
0154      CSR 10              Z-ADD1         E
0155      CSR 10    BDAY      SUB  1         D
0156      CSR       TAG11     TAG
0157      CSR       D         ADD  1         D
0158      CSR                 MOVE ET,E      EMT,D
0159      CSR       D         COMP METEST                71
0160      CSR 71              GOTO TAG12
0161      CSR       E         ADD  1         E
0162      CSR       E         COMP 28            29
0163      CSR 29    E         SUB  28        E
0164      CSR                 GOTO TAG11
0165      CSR       TAG12     TAG
0166      CSRN10    WORK      DIV  33        HASH
0167      CSRN10              MVR            I       20  33
0168      CSRN10 33           Z-ADD33        I           33
 RPG II COMPILER                                                    PAGE 0005

0169      CSRN10              Z-ADD0         D
0170      CSR 10              Z-ADD1         I
0171      CSR 10    BDAY      SUB  1         D
0172      CSR       TAG13     TAG
0173      CSR       D         ADD  1         D
0174      CSR                 MOVE IT,I      INT,D
0175      CSR       D         COMP METEST                71
0176      CSR 71              GOTO TAG14
0177      CSR       I         ADD  1         I
0178      CSR       I         COMP 33            34
0179      CSR 34    I         SUB  33        I
0180      CSR                 GOTO TAG13
```

```
0181      CSR         TAG14     TAG
0182      CSR                   Z-ADD1        X       20    78
0183      CSR                   Z-ADD0        D
0184      CSR         TAG15     TAG
0185      CSR         D         ADD  1        D
0186      CSR                   MOVELPHY,D    TL      20
0187      CSR                   MOVE PHY,D    TR      20
0188      CSR         TL        COMP 17                     77
0189      CSRN77      TR        COMP 17                     77
0190      CSR 77                MOVE D        CD,X
0191      CSR 77                MOVE 'P  '    TYPE,X
0192      CSR 77                SETON                 78
0193      CSR                   MOVELEMT,D    TL
0194      CSR                   MOVE EMT,D    TR
0195      CSR         TL        COMP 17                     77
0196      CSRN77      TR        COMP 17                     77
0197      CSR 77N78             MOVE D        CD,X
0198      CSR 77                MOVE 'E  '    TYPE,X
0199      CSR 77N78             SETON                 78
0200      CSR                   MOVELINT,D    TL
0201      CSR                   MOVE INT,D    TR
0202      CSR         TL        COMP 17                     77
0203      CSRN77      TR        COMP 17                     77
0204      CSR 77N78             MOVE D        CD,X
0205      CSR 77                MOVE 'I'      TYPE,X
0206      CSR 77N78             SETON                 78
0207      CSR 78      X         ADD  1        X             78
0208      CSR         D         COMP METEST                 72
0209      CSRN72                GOTO TAG15
0210      CSR                   SETON                       50
 RPG II COMPILER                                                    PAGE 0006

0211      CSR                   Z-ADD0        LN      20
0212      CSR         TAG16     TAG
0213      CSR         LN        ADD  1        LN
0214      CSR                   Z-ADD0        P
0215      CSR                   Z-ADD0        E
0216      CSR                   Z-ADD0        I
0217      CSR         TAG17     TAG
0218      CSR         P         ADD  1        P
0219      CSR         P         MULT 2        R       20
0220      CSR         R         SUB  1        L       20
0221      CSR                   MOVELPHY,P    PL      20
0222      CSR                   MOVE PHY,P    PR      20
0223      CSR         PL        COMP LN                     11
0224      CSR         PR        COMP LN                     12
0225      CSR 11                MOVE 'P'      LIN,L
0226      CSR 12                MOVE 'P'      LIN,R
0227      CSR         P         COMP METFST                 72
0228      CSRN72                GOTO TAG17
0229      CSR         TAG18     TAG
0230      CSR         E         ADD  1        E
0231      CSR         E         MULT 2        R
0232      CSR         R         SUB  1        L
0233      CSR                   MOVELEMT,E    EL      20
0234      CSR                   MOVE EMT,E    ER      20
0235      CSR         EL        COMP LN                     11
0236      CSR         ER        COMP LN                     12
0237      CSR 11      LIN,L     COMP ' '                    99
0238      CSR 11 99             MOVE 'E'      LIN,L
0239      CSR 11N99             MOVE '*'      LIN,L
0240      CSR 12      LIN,R     COMP ' '                    98
0241      CSR 12 98             MOVE 'E'      LIN,R
0242      CSR 12N98             MOVE '*'      LIN,R
0243      CSR         E         COMP METEST                 72
0244      CSRN72                GOTO TAG18
0245      CSR         TAG19     TAG
0246      CSR         I         ADD  1        I
0247      CSR         I         MULT 2        R
0248      CSR         R         SUB  1        L
0249      CSR                   MOVELINT,I    IL      20
0250      CSR                   MOVE INT,I    IR      20
0251      CSR         IL        COMP LN                     11
0252      CSR         IR        COMP LN                     12
 RPG II COMPILER                                                    PAGE 0007

0253      CSR 11      LIN,L     COMP ' '                    99
0254      CSR 11 99             MOVE 'I'      LIN,L
0255      CSR 11N99   LIN,L     COMP '*'                    87
0256      CSR 11N99N87          MOVE '*'      LIN,L
0257      CSR 11N99 87          MOVE '*'      LIN,L
0258      CSR 12      LIN,R     COMP ' '                    98
0259      CSR 12 98             MOVE 'I'      LIN,R
0260      CSR 12N98   LIN,R     COMP '*'                    83
0261      CSR 12N98N83          MOVE '*'      LIN,R
0262      CSR 12N98 83          MOVE '*'      LIN,R
0263      CSR         I         COMP METEST                 72
0264      CSRN72                GOTO TAG19
0265      CSR         LN        COMP 17                     17
0266      CSRN17                GOTO TAG21
0267      CSR                   Z-ADD0        Y       20
0268      CSR         TAG20     TAG
0269      CSR         Y         ADD  1        Y
0270      CSR         LIN,Y     COMP ' '                    18
0271      CSR 18                MOVE '-'      LIN,Y
0272      CSR         Y         COMP 62                     62
0273      CSRN62                GOTO TAG20
0274      CSR         TAG21     TAG
0275      CSR                   SETON                       51
0276      CSR         LN        COMP 33                     30
0277      CSR         LN        COMP 02                     41
0278      CSR         LN        COMP 04                     42
```

```
       CSR         LN       COMP 06                   43
       CSR         LN       COMP 08                   44
       CSR         LN       COMP 10                   45
       CSR 19      X        ADD  1       X
       CSR 19      X        COMP 9                 19
       CSR                  EXCPT
       CSR 17               SETON                19
       CSR 17               Z-ADD0       X
       CSR                  SETOF                5051
       CSRN30               GOTO TAG16
       CSR         THONTH   ADD  1       THONTH
       CSR         THONTH   COMP XMONTH               97
       CSR 97               GOTO TAG23
       CSR 10      2        SUB  BDAY    WORK
       CSR 10               SETOF                10
       CSR         WORK     ADD  METEST  WORK
 RPG II COMPILER                                                    PAGE 0008

       CSR         C        ADD  1       C
       CSR         C        COMP 12              13
       CSRN13               GOTO TAG22
       CSR         C        SUB  12      C
       CSR         CYEAR    ADD  1       CYEAR
       CSR         CLY      ADD  1       CLY          89
       CSRN89      CYEAR    DIV  4       TEST
       CSRN89               MVR          LEAP         94
       CSR 89      CYEAR    DIV  400     TEST
       CSR 89               MVR          LEAP         94
       CSR         TAG22    TAG
       CSR                  MOVE C       CHONTH
       CSR 94      CHONTH   COMP 2                    92
       CSR                  GOTO TAG08
       CSR         TAG23    ENDSR
       OREPORT  E  205   50
       O                          MON.C     4
       O                          CYEAR     9
       O                                   35 'BIORHYTHM CHART FOR '
       O                          NAME     67
       O                                   73 ' BORN '
       O                          MON.B    77
       O                          BDAY     80
       O                                   82 ', '
       O                          BYEAR    86
       O        E  1     50
       O                                   34 '01  03  05  07  09  11  '
       O                                   58 '13  15  17  19  21  23  '
       O                                   72 '25  27  29  31'
       O        E  1     50
       O                                   34 '  02  04  06  08  10  12'
       O                                   58 '  14  16  18  20  22  24'
       O                                   72 '  26  28  30  '
       O        E  1     50
       O                                   33 '************************'
       O                                   57 '************************'
       O                                   73 '****************'
       O        E  1     51
       O                                   10 '*'
       O                          LIN    B 72
       O                                   73 '*'
       O                 41                90 'PPPP - PHYSICAL'
 RPG II COMPILER                                                    PAGE 0009

       O                 42                92 'EEEE - E EMOTIONAL'
       O                 43                94 'IIII - INTELLECTUAL'
       O                 44                96 '**** - 2 CYCLES CROSS'
       O                 45                96 '**** - 3 CYCLES CROSS'
       O                 17                87 'CRITICAL DAYS'
       O                 19       CD.X   Z8 80
       O                 19       TYPE.X B 85
       O        E      01 30
       O                                   33 '************************'
       O                                   57 '************************'
       O                                   73 '****************'
       O        E   01   02
       O       OR 1      03
       O       OR 2      04
       O       OR 3      05
       O                          LINE     75
 **
 JAN.FEB.MAR.APR.MAY JUNEJULYAUG.SEP.OCT.NOV.DEC.
 **
 312831303130313130313031
 **
 171615141312111009080706060708091011121314151617181920212223242526272828272625 24
 232221201918
 **
 1716151413121110090807060504030405060708091011121314151617181920212223242526272 8
 2930313029282726252423222 1201918
 **
 17161514131211100908070605040302010102030405060708091011121314151617181920212223
 242526272829303132333332313029282726252423222 1201918
 **
```

```
                               BIORHYTHMIC                             1
I.    THEORY                                                           4
      BIORHYTHMIC THEORY HOLDS THAT FROM THE MOMENT OF BIRTH. OUR PHYSICAL. 2
      EMOTIONAL AND INTELLECTUAL ENERGIES CAN BE CHARTED AS FIXED.     2
      REPETITIVE CYCLES.                                               2
      A.   PHYSICAL CYCLE                                              3
           THIS IS THE SHORTEST CYCLE. BEING 23 DAYS IN DURATION.  IT IS 2
           SAID TO AFFECT THE PHYSICAL CHARACTERISTICCS OF MAN - STRENGTH. 2
           ENERGY. ENDURANCE. RESISTANCE. ETC.                         2
      B.   EMOTIONAL CYCLE                                             2
           THIS CYCLE IS 28 DAYS LONG AND CONTROLS THE STATE OF MAN'S  2
           EMOTIONS. MOODS. SENSITIVITY. MENTAL STABILITY AND CREATIVITY.  2
```



```
      C.   INTELLECTUAL CYCLE                                          2
           THIS LONGEST CYCLE. BEING 33 DAYS IN LENGTH. CONTROLS THE   2
           MEMORY. CONCENTRATION. RESPONSIVENESS. AND OTHER FUNCTIONS OF 2
           THE MIND.                                                   2
      EACH CYCLY IS DIVIDED INTO TWO PARTS.  THE FIRST HALF IS USUALLY 3
      CALLED THE HIGH CYCLE. ALSO ACTIVE OR POSITIVE CYCLE.  DURING THIS 2
      HALF OF THE CYCLE. THE PHASES ARE SAID TO BE DISCHARGING.  IF IT IS 2
      THE PHYSICAL CYCLE. WE ARE ACTIVE. EMOTIONAL. MORE ENTHUSIASTIC. 2
      INTELLECTUAL. ABLE TO ABSORB NEW MATERIAL.  IN THE SECOND HALF. KNOWN 2
      AS THE LOW. PASSIVE OR NEGATIVE PHASE. THE CYCLES ARE RECHARGING.  2
      EACH CYCLE TAKES THE OPPOSITE CHARACTERISTIC OF THE HIGH PHASE.  THIS 2
      IS NOT A BAD PERIOD IN THE CYCLES.  SINCE THE CYCLES ARE DIFFERENT  2
      LENGTHS. THEY WILL BE IN DIFFERENT PHASES ON A GIVEN DATE OR THE CYCLE2
      WILL BE ON THE LINE BETWEEN PHASES.  THIS OCCURS TWICE. ON THE FIRST 2
      DAY AND IN THE MIDDLE OF EACH CYCLE.  THESE DAYS ARE CALLED THE  2
      CRITICAL DAYS.  IT IS ON THESE DAYS THAT THE CYCLE IS PASSING FROM 2
      POSITIVE TO NEGATIVE OR VICE VERSA AND THE CYCLE IS IN FLUX.  ON 2
      THESE DAYS. OUR REACTIONS TO EVENTS AROUND US MAY NOT BE NORMAL. 2
      INDUSTRIAL STUDIES HAVE SHOWN THAT 60% OF INDUSTRIAL ACCIDENTS   3
      OCCURRED ON CRITICAL DAYS. ALTHOUGH CRITICAL DAYS REPRESENT ONLY 2
      20% OF OUR DAYS.  IF ACCIDENTS WERE EVENLY DISTRIBUTED THROUGHOUT  2
      OUR LIVES. ONLY 20% SHOULD OCCUR ON CRITICAL DAYS.  IN OTHER WORDS. IT2
      IS THREE TIMES MORE LIKELY FOR AN ACCIDENT TO OCCUR ON A CRITICAL DAY 2
      AS ON A NON-CRITICAL DAY.                                        2
II.   HISTORY                                                          4
      THOUGH WORKING INDEPENDENTLY. DR. HERMANN SWOBODA. A PROFESSOR OF 2
      PSYCHOLOGY AT THE UNIVERSITY OF VIENNA. AND DR. WILHELM FLIESS. A 2
      NOSE AND THROAT SPECIALIST IN BERLIN. DISCOVERED THE PHYSICAL AND 2
      EMOTIONAL CYCLES.  IN THE 1920S. ALFRED TELTSCHER. AUSTRIAN DOCTOR 2
      OF ENGINEERING IN INNSBRUCK. DISCOVERED THE INTELLECTUAL CYCLE IN 2
      HIS OBSERVANCE OF HIS STUDENTS' DAILY ABILITY TO ABSORB MATERIAL. 2
      ALTHOUGH THE THEORY HAS BEEN ACCEPTED FOR THE PAST FIFTY YEARS. IT 2
      HAS NOT HELD WIDE-SPREAD USE BECAUSE IT IS VERY TIME CONSUMING TO 2
      CALCULATE THE THREE CYCLES.  HOWEVER. WITH THE USE OF COMPUTERS. 2
      MANY COMPANIES ARE NOW CHARTING THEIR EMPLOYEES' CRITICAL DAYS TO  2
      ADVISE THEM TO BE MORE CAREFUL ON THOSE DAYS.  UNITED AIRLINES'  2
      GROUND CREW AT NATIONAL AIRPORT IN WASHINGTON. D.C. CUT ACCIDENTS 2
      BY HALF OVER A ONE YEAR PERIOD AFTER THEIR CYCLES WERE CHARTED.  A  2
      BUS COMPANY IN TOKYO. JAPAN HAS REPORTED A 50% REDUCTION IN ACCIDENTS.2
III.  USES                                                             1
      THE FIRST USE. OBVIOULSY. IS TO DETERMINE THE CRITICAL DAYS IN EACH  2
      MONTH AND TO TAKE CARE TO BE MORE CAREFUL THEN  IF ANY DAY IS A  2
```



```
      DOUBLE OR TRIPLE CRITICAL DAY. THAT IS. TWO OR THREE CYCLES CROSSING  2
      THE LINE AT THE SAME TIME. ADDITIONAL CARE SHOULD BE TAKEN.      2
      SECOND. WE CAN OBSERVE HOW THE THREE CYCLES FALL DURING THE MONTH. 3
      HIGH OR LOW.  WHEN POSSIBLE. USE THE HIGH PHASE TO OUR ADVANTAGE AND 2
      BE AWARE OF THE LOW PHASES.  FOR EXAMPLE. THE LOW PHASE OF THE   2
      INTELLECTUAL CYCLE MAY NOT BE THE BEST TIME TO START A NEW PROJECT.  2
      IT HAS BEEN OBSERVED THAT BABIES CONCEIVED DURNING A PHYSICAL HIGH 3
      WITH THE EMOTIONAL CYCLE LOW HAVE USUALLY BEEN BOYS.  WHEN THE   2
      OPPOSITE OCCURRED. EMOTIONAL CYCLE HIGH AND PHYSICAL LOW. IT HAS 2
      BEEN A GIRL.  BOTH HIGH OR LOW HAS BEEN UNPREDICTABLE.           2
      HOWEVER. BIORHYTHM CHARTS ARE ONLY A GUIDE AS TO HOW YOUR CYCLES MAY 3
      BE WORKING.  IT IS NOT PREDICTING FACT.  THEY MIGHT BE COMPARED TO A 2
      ROAD MAP.  IT SHOWS YOU HOW TO GET TO A GIVEN POINT BUT CANNOT   2
      PREDICT WHAT WILL ACTUALLY HAPPEN IF YOU TAKE A GIVEN ROUTE ALONG 2
      THE WAY.                                                         2
IV.   ANALYSIS OF THE CHART                                            4
      ON THE RIGHT HAND SIDE OF THE CHART. THERE IS A LEGEND TO DENOTE THE  2
      THREE CYCLES ON THE GRAPH.  IF THE CYCLES GROSS OR ARE IN COMPLETE  2
      PHASE WITH ONE ANOTHER. A '+' OR '*' IS USED.  BELOW THAT. THE   2
      CRITICAL DAYS FOR THE MONTH ARE LISTED WITH A LETTER. P FOR PHYSICAL. 2
      E FOR EMOTIONAL. OR I FOR INTELLECTUAL. TO THE RIGHT.  THIS TELLS YOU 2
      WHAT CYCLE IS CRITICAL THAT DAY.  IF THERE ARE TWO OR THREE LETTERS  2
      THERE. THIS IS A DOUBLE OR TRIPLE CRITICAL DAY.  REVIEW THE CHART TO  2
      DETERMINE YOUR HIGH AND LOW PHASES FOR THE MONTH                 2
                                                                       1
/*

THE FOLLOWING INDICATORS APPEARED IN THIS PROGRAM

01  02  03  04  05  06  07  08  09  10  11  12  13  17  18  19
23  24  28  29  30  33  34  41  42  43  44  45  50  51  62  71
72  77  78  81  82  83  85  87  88  89  92  94  95  96  97  98
99
```

SORTED LABEL NAMES

NAME	LINE
CHRTSR	68
DOC#1	54
DOCSR	51
END	50
TAG01	84
TAG02	89
TAG03	100
TAG04	103
TAG05	116
TAG06	118
TAG07	126
TAG08	128
TAG09	140
TAG10	149
TAG11	156
TAG12	165
TAG13	172
TAG14	181
TAG15	184
TAG16	212
TAG17	217
TAG18	229
TAG19	245
TAG20	268
TAG21	274
TAG22	305
TAG23	309

RPG II COMPILER PAGE 0012

SORTED FIELD NAMES

NAME	TYPE	LENGTH	DP
B	N	2	0
BDAY	N	2	0
BLY	N	2	0
BMONTH	N	2	0
BYEAR	N	4	0
C	N	2	0
CD	N	2	0
CLY	N	2	0

RPG II COMPILER PAGE 0013

NAME	TYPE	LENGTH	DP
CMONTH	N	2	0
CYEAR	N	4	0
D	N	2	0
DOC	A	76	
E	N	2	0
EL	N	2	0
EMT	A	4	
ER	N	2	0
ET	A	4	
HASH	N	5	0
I	N	2	0
IL	N	2	0
INT	A	4	
IR	N	2	0
IT	A	4	
L	N	2	0
LEAP	N	4	0
LIN	A	1	
LINE	A	75	
LN	N	2	0
M	N	2	0
MCNT	N	2	0
METEST	N	2	0
MON	A	4	
NAME	A	32	
P	N	2	0
PHY	A	4	
PL	N	2	0
PR	N	2	0
PT	A	4	
R	N	2	0
TEST	N	4	0
TL	N	2	0
TLY	N	2	0
TMONTH	N	4	0
TR	N	2	0
TYEAR	N	4	0
TYPE	A	3	
UDATE	N	6	0
UDAY	N	2	0
UMONTH	N	2	0
UYEAR	N	2	0

RPG II COMPILER PAGE 0014

NAME	TYPE	LENGTH	DP
WORK	N	5	0
X	N	2	0
XMONTH	N	4	0
Y	N	2	0
YZ	N	2	0
ZY	N	1	0

Macro assembler format consists of four fields: the location field, the operation field, the address field, and the comments field.

Location	Operation	Address	Comments

The total width of all four fields combined is 72 columns. Each field can be any length. A blank signals the end of a field. The next nonblank character begins the next field. An asterisk in column 1 indicates a comment statement. Pressing LINE FEED, RETURN on the operator's console signifies the end of a statement entered from the console. The end of the card signifies the end of a statement for card input.

INDEX

COMMENT SHEET

MANUAL TITLE CDC® MSOS 5 Version 5 Installation Handbook

PUBLICATION NO. 96769410 REVISION A

FROM NAME:

BUSINESS
ADDRESS:

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

CUT ALONG LINE

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 333

LA JOLLA, CA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

CUT ALONG LINE

FOLD

STAPLE

STAPLE

CORPORATE HEADQUARTERS, P O BOX 0, MINNEAPOLIS, MINNESOTA 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A

CONTROL DATA CORPORATION